TEL AVIV UNIVERSITY

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING

The Zandman-Slaner Graduate School of Engineering

TRANSONIC FLOW COMPUTATIONS BY AN ALGEBRAIC MULTIGRID METHOD

By

Shlomy Shitrit

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY

in partial fulfillment of the requirement for the degree of

"DOCTOR OF PHILOSOPHY"

September 2010

TEL AVIV UNIVERSITY

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING

The Zandman-Slaner Graduate School of Engineering

TRANSONIC FLOW COMPUTATIONS BY AN ALGEBRAIC MULTIGRID METHOD

By

Shlomy Shitrit

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY

in partial fulfillment of the requirement for the degree of

"DOCTOR OF PHILOSOPHY"

Under the Supervision of Prof. Alexander Gelfgat and Dr. David Sidilkover

September 2010

Acknowledgments

First and Foremost I would like to personally express my gratitude to my supervisor Dr. David Sidilkover for his supervision and guidance throughout the course of this study. I thank him for the interesting and stimulating discussions we had and for the encouragement he has given me in the completion of this thesis. Thanks to his profound knowledge, instructions, and insights into CFD and numerical analysis, I have not only learned a great deal about the subject, but have come to enjoy this field of research. I would also like to thank Professor Alexander Gelfgat for his guidance and for the time that he has invested in me as well. I would like to thank Dr. Shlomit Gali and Tal Joshua for supporting this research work and being a constant source of encouragement.

Last, but certainly not least, I would like to thank my beloved wife Chen. You are my eternal light, guiding and supporting me, even in times of complete darkness.

Abstract

The objective of this work is to develop a highly efficient solver for the Full Potential Equation (FPE) that will be able to compute transonic external and internal flows attaining a (nearly) linear computational complexity. The key innovation of this work is in the solver's efficiency and in the fact that it is achieved by means of adapting and applying the algebraic multigrid (AMG) approach to solving the problem. The mathematical difficulties of the problem are associated with the fact that the governing equation changes its type from elliptic (subsonic flow) to hyperbolic (supersonic flow). A pointwise relaxation method when applied directly to the upwind discrete operator, in the supersonic flow regime, is unstable. Resolving this difficulty is the main achievement of this work. A stable pointwise direction-independent relaxation was developed for the supersonic and subsonic flow regimes. This stable relaxation is obtained by post-multiplying the original operator by a certain simple first order downwind operator. This new operator is designed in such a way that the pointwise relaxation applied to the product operator becomes stable. A variety of issues regarding the AMG coarsening and construction of transfer operators is addressed in order to achieve the required efficiency for the problems under consideration.

An improved coarsening process was developed. Instead of using a fixed threshold parameter in order to select the coarse-grid points, we developed a dynamic threshold parameter as a measure of the strength of connection between the matrix variables. The coarsening by dynamic threshold was shown to be less effective for certain elliptic problems (subsonic flow), but for supersonic flow regime where the operator does not form an M-matrix, we obtained much better performance. In some cases where an irregular grid, shock waves, and extreme nonlinearity are involved, the dynamic threshold is more than necessary in order to achieve convergence. A modified formulation of the interpolation operator is presented. While the standard interpolation is suitable mainly for problems that are characterized by M-matrix form, the proposed formula is more accurate and can be used for more general matrix problems. The proposed interpolation operator includes the choice of negative weights, which is necessary in some cases. In addition, the FMG approach in the context of AMG was developed as a tool to deal with a nonlinear problems. This approach significantly improved the initial condition, especially in nonlinear cases where the initial approximation is critical, to start the iterative procedure from a good initial approximation.

In the first part of this work we demonstrate the AMG performance on a variety of model problems involving the quasi-linear full potential equation in two dimensions. Finite difference methods were applied for subsonic, sonic, and supersonic flow for various flow directions (with respect to the grid). In addition, the capabilities of the constructed algorithm to deal with a nonlinear problem - a transonic small-disturbance equation – is presented. The second part of the work includes the solution of the FPE in the conservation form. A FPE solver was developed, based on the AMG method. The flow solver is capable of flow from subsonic to transonic conditions. Several two dimensional flow calculations have been performed to test the performance of the algebraic multigrid method implemented on the FPE under body-fitted structured grid configurations. If lift is being produced, the Kutta-Joukowski condition is enforced for the circulation. In order to present the algorithm's performance, results of internal and external flows

are presented for subsonic and transonic flows. The proposed algorithm is demonstrated to attain excellent convergence rates, independent of the problem size, through the entire range of flow regimes. Finally, the 2D stable operator in the supersonic flow regime was extended to 3D. We present a 3D pointwise relaxation procedure that is stable in both the subsonic and supersonic flow regimes. This was verified by the Von-Neumann stability analysis.

Contents

Nomenclature

a	= speed of sound; matrix entry
M	= Mach number; transformation matrix
R	= specific gas constant
r	= residual
ϕ	= potential function
u, v, w	= Cartesian velocities in x, y and z directions
q	= magnitude of the velocity
$\hat{n}$	= unit normal vector
h	= grid mesh size, enthalpy per unit mass
G	= amplification factor
$\overrightarrow{n}$	= normal vector
Q	= source term
f	= flux function
e	= internal energy per unit mass
T	= temperature
p	= pressure
ρ	= density; spectral radius
D	= diagonal matrix
L,U	= lower and upper matrices
s	= entropy
C_p	= specific heat at constant pressure
C_v	= specific heat at constant volume
Γ	= circulation
ω	= under relaxation parameter
$\theta_1, \theta_2, \theta_3$	= Fourier frequencies in x, y and z directions
L	= operator
e	= algebraic error
I_m^{2m}	= restriction operator

I_{2m}^{m} = interpolation operator

Ω = domain of the PDE

f = equation right hand side

ν = number of relaxations before and after visiting the coarse-grid

μ = coarse-level matrix reduction parameter

n = number of points in a given direction

WU = work units

ε = threshold parameter

S = strength matrix; face vector

V = velocity

k = thermal conductivity

H = enthalpy

C_f = convergence factor

C_g = grid complexity

ξ, η = co-variant coordinate system

ξ_n, η_n = contra-variant coordinate system

$\triangle S$ = face length

Subscripts

i, j, k= identification of each cell center

∞ = free stream value

x = derivative with respect to x

y = derivative with respect to y

s = derivative with respect to s

n = derivative with respect to n

Superscripts

m = coarse-level

n = iteration index

d = dimension of problem

Abbreviations

AMG = algebraic multigrid

CFD = computational fluid dynamics

FPE = full potential equation

PDE = partial differential equation

TSD = transonic small disturbance

FMG = full multigrid

FAS = full approximation scheme

DNS = direct numerical simulation

LES = large-eddy simulation

RANS = Reynolds average Navier-Stokes

NACA = national advisory committee

List of Figures

List of Tables

1 Introduction

1.1 About this work

The objective of this work is to develop a highly efficient solver for the Full Potential Equation (FPE), which will be able to compute transonic external and internal flows attaining a (nearly) linear computational complexity.

The key innovation of this work is in the solver's efficiency and the fact that it is achieved by means of adapting and applying the algebraic multigrid (AMG) approach to solving the problem.

The practical purpose of the proposed development is twofold:

- A standalone FPE solver.
- A building block for the compressible Euler and Navier-Stokes solver, based on the *factorizable* discretization methods (see the explanations below).

1.2 Historical and scientific background

The complete Navier-Stokes equations are considered to be the correct mathematical description of the governing equation of fluid motion. The system of equations describes the conservation of mass, momentum, and energy, and is highly coupled and nonlinear. Simplifications to the Navier-Stokes equations are made whenever possible. For certain types of flows, these approximations can be made without much compromising of the physical model. Some of the various approximations will now be discussed.

Solutions to the Navier-Stokes equations for laminar flow are considered to be as accurate as numerical computations can be. But, most problems where viscous effects are important are classified as turbulent flows. The Direct Numerical Simulations (DNS) using Navier-Stokes equations can be performed to resolve the small scale structure of turbulent flow, but the problem is with the enormous number of grid points required to capture the physics of the flow. Another category of turbulence modeling is known as large eddy simulation (LES). In this approach the large scale turbulent motion is solved by refining the grid. The small scale is solved by employing a turbulence model. The viewpoint behind this theory is that the large turbulent motion is characterized by the boundary conditions, and should not be modeled, but rather computed. On the other hand, the smaller turbulent motions are more isotropic, which can therefore be resolved using a general model. The LES methodology was employed already in 1963 by Smagorinsky [1]. The first engineering application of LES was presented by Deardorff in 1970 [2]. The next level of approximation is the Reynolds-Averaged Navier Stokes (RANS) equations [3, 4]. Some examples of where the RANS equations are applied are flow around a 3D obstacle [5, 6], flow in an internal combustion engine [7], unsteady oscillatory flow in an inlet [8, 9], and flow in a turbine blade row [10].

Another approximation depends on how the density varies throughout the flow field. For many cases of flow that occur naturally, the variance in density is negligible compared to the other flow parameters. For example, in air the density can be assumed constant without any significant loss of engineering accuracy for Mach numbers below 0.3 [11, 12].

The next level of approximation is when the viscous layers are thin. This situation describes flow with a small amount of separation or back-flow where viscous and turbulent diffusion can be neglected in the mainstream direction. The Euler equations describing inviscid flow are obtained when the viscous terms in the Navier-Stokes equations are neglected. Since the viscous terms no longer need to be involved in the solution process, and the grid does not need to be clustered near the wall to resolve the boundary layer, the computational cost is significantly lower. An example of where the thin shear layer approximation is applied is viscous flow around an airfoil [13].

The next approximation under consideration is the potential equation [12, 14, 15], which describes irrotational flows. In this inviscid irrotational flow, the entropy will be constant over the whole flow field. A flow with constant entropy is known as isentropic flow. A set of isentropic relations becomes basis of simplifying the Euler equations. The potential flow model is equivalent to the Euler equations for continuous, irrotational flows. For subsonic external and internal flows, the solution to the Euler and potential equations are in many cases almost identical [12, 16, 17, 18, 19]. The difference between the solutions of the two models become more evident for supersonic flows with shock waves. The main advantage in the potential flow model, whenever adequate, is that there is only one equation to solve, instead of a set of five equations for the 3D Euler system. In spite of the limitations associated with the potential flow model, it is still useful in engineering applications. A robust and efficient solver for the FPE may still be preferable in many cases over a more complicated Euler equation solver, provided its computational efficiency is substantially higher. One such case is an aerodynamic design problem, when the flow field solution should be computed repeatedly many times with variations of the body geometry. An efficient solver, even if for a simplistic model, can be highly desirable in such a situation. This fact makes the potential flow model very valuable in the design process since the basic physics of an inviscid flow field is still captured.

A need for efficient and accurate solvers for compressible flow equations exists in many areas of engineering and science. A necessity to answer this need is what keeps motivating the development of such solvers for several decades starting from the beginning of the computer age. However, due to the number and complexity of the issues that need to be addressed, there still is much to be done. Much of the research on numerical methods for potential flow was conducted throughout the 1970s and into the early 1980s. One of the major early breakthroughs in this development was the work by Murman and Cole on numerical solution of the small disturbances equation for transonic flow [20]. The key achievement of the work was the realization that since the governing equation changes its type from elliptic to hyperbolic where the flow reaches supersonic speeds, this should be reflected in the discretization of the equation. Each mesh point was correctly treated using a type-dependent difference based on the domain of dependence. Their paper laid the ground work for the years that followed. In 1973, Murman presented a solution of the Transonic Small Disturbance (TSD) equation by using central differencing for the subsonic regions and upwind differencing for the supersonic region [21]. After the work of Murman and Cole other researchers kept developing this idea. Steger and Lomax [22] presented the successive over-relaxation scheme (SOR) to solve the potential equation. They presented transonic airfoil solutions.

Garabedian and Korn [23] increased the order of accuracy for the Murman and Cole's scheme and also solved the FPE in nonconservative form. In 1972 Ballhaus and Bailey [24] and Bailey and Steger [25] both solved transonic flows around wings using the TSD equation.

In 1974 the idea of Murman and Cole was extended to the nonlinear FPE and arbitrary flow direction by Jameson [26, 27]. The key issue was reflecting the domain and dependence when the flow was generally supersonic. Jameson came up with the idea of the rotated difference scheme using conformal mapping to solve two- and three- dimensional problems. In 1975, he presented a solution for the FPE in conservation form [28, 29, 30]. The concept of artificial viscosity was applied to the difference scheme in the supersonic region to achieve upwinding. In 1977, the first finite volume computations were presented by Jameson and Caughey [31, 32, 33]. They developed an approach where the fluxes at a cell face were obtained using averages from the corner points. An important result of their paper was that nonorthogonal curvilinear grids were used for dealing with complex boundaries.

In 1978 Hafez et al. [34] solved the FPE in conservation form by applying artificial compressibility. In this concept, the needed viscosity in the supersonic region was introduced by modifying the density in such a way that the upwinding bias would be introduced. This approach was called density biasing and the presented examples were flow around a cylinder and a NACA-0012 airfoil.

In 1978 various approaches were based on the approximate factorization method. These schemes were developed by Ballhaus and Jameson [35] and Baker [36]. A more complete description of the approximate factorization scheme is given by Holst and Ballhaus [37].

A multigrid method for convergence acceleration was applied first in 1977 by South and Brandt [38]. They solved the TSD equation. Other multigrid applications from that time were published by Holst [39]. A multigrid algorithm for the FPE in conservation form was first applied by Jameson in 1979 [40, 41]. The ADI iteration scheme in each coarse-level was applied. Different multigrid strategies for solution of the potential equation are presented by Van-Der Wees et al. in [42].

A more recent contribution to the potential flow research area has been made by Holst [43]. In 1995 he developed a potential flow solver, using the chimera grid approach, that can deal with three-dimensional geometries.

As can be seen, the time between the 1970s and 1980s was a period of rapid development of potential flow solvers. Contributions were made by numerous researchers. It seems, however, that this direction was abandoned while still being in its infancy following another groundbreaking work [44], in which a method for solving the Euler equations was devised. This methodology was generalized further to Navier-Stokes equations, turbulent flows, etc. and became the de facto standard still accepted by aerospace industries all over the world. However, it has been clear for some time (more than a decade, until few years ago) that only small incremental efficiency improvements were made in the framework of this approach. Only in the recent years a version of Runge-Kutta Implicit technique by was developed (see, for instance [45]) resulted in a substantial performance increase. In the opinion of numerous researchers, a substantial departure from the above methodology is required in order to facilitate further progress. One of such possible directions originates from the recommendation by Brandt [46]: since the

system of equations is of the mixed type, it is beneficial in terms of efficiency to address each of the co-factors in a way that is the most efficient for it instead of treating the whole system in the same way.

This idea was successfully realized in the past for the incompressible high Reynolds number Navier-Stokes equations (see, for instance, [47, 48]), but the progress towards applying it to the compressible flow was rather slow and the success is very limited. The explanation for this is in the complexity of the issues that need to be resolved. One such difficulty is that the standard discretization schemes in multidimensions introduce non-physical coupling between the different co-factors of the system. This difficulty is addressed by the emerging class of the so-called *factorisable* methods [49, 50]. In this light, the task of constructing an efficient FPE solver attains great importance, since such a solver can be used not only by itself, but becomes an integral part of the overall methodology for solving the flow equations based upon the factorizable discretization.

2 Research Objectives and Significance

The purpose of this work is to develop a highly efficient method for solving the FPE equation. The practical goal of this is two-fold: first, as was mentioned above, to develop an important building block for the factorizable methodology. Second, to develop a stand-alone "optimally" efficient transonic FPE solver. Such a solver can be a useful tool for engineers, for example, in the aerodynamic design process, where multiple flow field computations need to be performed as small changes to the geometry are made.

Remark. It appears that despite the availability of the Navier-Stokes codes, the engineers in aerospace industries would still often use an FPE solver, provided it is a really efficient one. They sometimes prefer simplistic flow solutions to a more meaningful Navier-Stokes solution if the former can be obtained cheaply (with minimal time and with modest computer resources). For instance, a group of researchers developed a method for solving FPE which appeared significantly more efficient than any of its predecessors about a decade ago (see [51]). This methodology was soon adopted by some aerospace industries (such as the Boeing company), where it is still used routinely (as far as we know).

Multigrid methods are considered to be most efficient for a variety of problems. In 1962, Fedorenko [52, 53] introduced the first instance of a class of algorithms that would come to be known as multigrid methods [54, 55, 56, 57, 58, 59]. Fedorenko solved the Poisson equation in a unit square. Since then, other mathematicians have extended Fedorenko's idea to general elliptic boundary value problems with variable coefficients; see, e.g., [60]. However, the full efficiency of the multigrid approach was realized by Brandt [61, 62, 63]. He also made these methods applicable to nonlinear problems by introducing the Full Approximation Scheme (FAS) [62]. Another achievement in the formulation of multigrid methods was the full multigrid (FMG) scheme [62], based on the combination of nested iteration techniques and multigrid methods. For additional references of multigrid methods see, e.g., [64, 65, 66].The growing demand for efficient solvers led to further developments of geometric multigrid methods. However, for many applications it is difficult to construct a sequence of coarse-grids. Furthermore, geometric multigrid

methods are, in general, not robust with respect to the operator coefficients. General anisotropies make an application of the geometric multigrid very cumbersome. In the 1980s algebraic multigrid (AMG) methods were developed [67, 68, 69, 70] to deal with these problems by extending the main ideas of geometric multigrid methods to an algebraic setting. The AMG is a method for solving algebraic systems based on multigrid principles with no explicit knowledge of the problem geometry. AMG uses the matrix's properties to construct the coarse-levels and the operators involved in the algorithm.

This work concentrates on applying the algebraic multigrid approach to computational aerodynamics problems, with the purpose of obtaining robust and efficient solver methodology suitable for practical purposes. A basic problem of computational aerodynamics - compressible Euler equations - can be characterized by an inherent anisotropy, attributed to the scalar differential operators of which the system of PDEs is comprised,

- advection operator,
- full-potential operator at nearly sonic flow speeds.

Having to deal with these general anisotropies makes an application of the "traditional" geometric multigrid very complicated [71, 72]. The algebraic multigrid approach, however, addresses this issue for a scalar operator case in a natural way. Applying these methods to systems of equations, though, is a difficult task. This is due to the fact that different kinds of anisotropy can be found in the physical phenomena described by the system. The factorizable methods, which have been demonstrated to have certain advantages in facilitation of optimal solvers, allow addressing each scalar operator separately, possibly by algebraic multigrid.

This work investigates the possibility of applying an algebraic multigrid method to transonic flow problems. The mathematical difficulties of the problem are associated primarily with the fact that the governing equation changes its type, being elliptic in the subsonic and hyperbolic in the supersonic regions of the flow. Since these two cases differ in their properties, a suitable numerical approximation should be devised for each of these two regions. It was first shown by Murmann and Cole in 1971 [20] that the stable solution to the transonic small disturbances equation can be obtained by switching from central differencing in the subsonic region to upwind differencing in the supersonic region and applying the line implicit relaxation scheme [38]. This methodology was generalized by Jameson to the FPE and general flow direction [73, 74].

2.1 Thesis outline

This thesis can be divided into two parts. The first, including Section 3 to Section 9) describes the basic AMG method and the extension of the method to deal with transonic flow problems that are characterized by strong nonlinearity, shock waves, and extreme anisotropy (results of the irregularity of the grid or the equation itself). The second part (from Section 10 to Section 12) deals with the FPE in the quasi-linear form solved with finite difference approach. The third part describes the solution of the FPE in the conservation form while the equation is discretized on a body-fitted structured grid approach. Now these sections are described in detail:

First part

Background on the classical geometric multigrid method is given in Section three. The full multigrid as a tool to improve the initial guess is also discussed in this section. In Section four the basic AMG method is described. Details regarding computational complexity issues and the advantages and difficulties in using a pointwise relaxation method are given there as well. Relaxation methods and smoothing analysis are described in Section 5. The transonic flow problem is introduced in Section 6 and our approach to solve the supersonic region is presented. In Section 7, the extensions of the AMG method to the transonic flow computations are discussed. Numerical calculations using finite difference method are presented in Section 8 and convergence results for various flow speeds are given. The ability of the developed AMG approach to deal with the nonlinear TSD equation is presented in Section 9.

Second part

A very brief introduction to finite volume method is presented in Section 10. The discretization of the FPE is discussed in Section 11. In Section 12, we present the results of several numerical experiments with the AMG method. Here, we also consider discretizations on body-fitted structured grids. The feasibility study of the developed methodology's extension to a three-dimensional case is presented in Section 13. Finally we conclude with some remarks in Section 14.

3 Geometric (Classic) Multigrid Method

The multigrid method is a multilevel iterative method for solving the algebraic system $Au = f$, where A is a real $n \times n$ matrix and u, f are vectors in $\mathbb{R}^n$. We denote by v an approximation of the exact solution u and by e the error, $e = u - v$. Defining the residual to be $r = f - Av$, we observe the critical relationship known as the residual equation $Ae = r$. Multigrid is the recursive application of a two-grid process.

An iterative method, such as Jacobi or Gauss–Seidel, which is applied to the fine-grid problem, is characterized by a slow convergence rate that also depends on the grid size $(1 - O(h^2))$. This method provides rapid damping for the high frequency error components, while having very little effect on the smooth low frequency errors. These smooth components are responsible for the slow total convergence. Consider, for example, performing the Jacobi iteration on the 2D finite difference Laplacian on a 50×50 grid, as shown in Figure 1. This behavior is typical of stationary iterations applied to elliptic operators: oscillatory errors reduce quickly, while the slow convergence is seen in smooth modes.

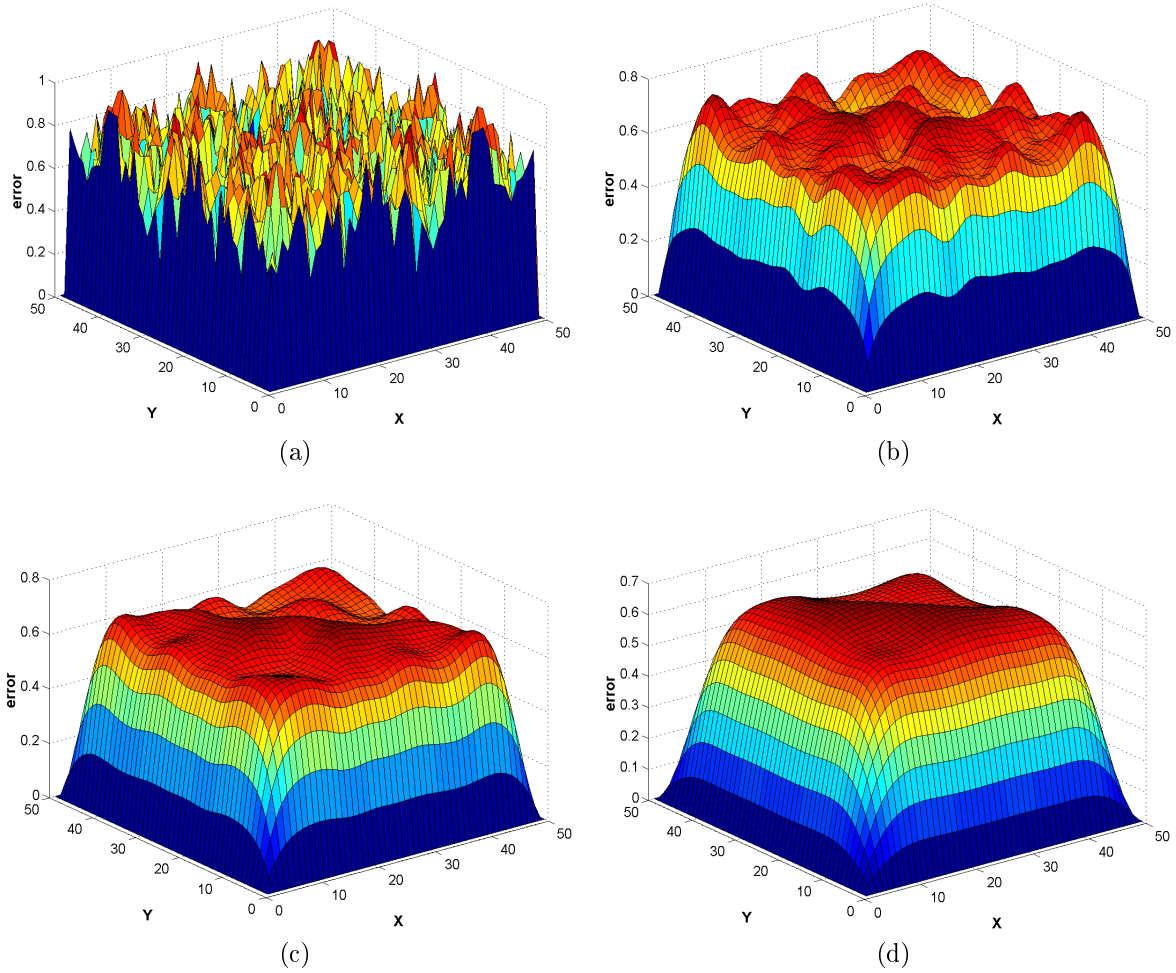


Figure 1: Smoothing of a random error by the Jacobi iteration method.

A multigrid algorithm, employing grids of different mesh size, allows solving for all the error components and provides a rapid convergence rate. The high frequency components are reduced by applying a pointwise relaxation method like Jacobi or Gauss–Seidel, while the low frequency error components are effectively reduced by a coarse-grid correction procedure. Therefore, in order to effectively reduce all error components, it seems appropriate to use the recursive error reduction method, and transfer the smooth error components several times to successively coarser grids to allow for effective error reduction. This idea leads to the concept known in multigrid as V-cycle. It is illustrated in Figure 2. The fundamental idea behind the multigrid algorithm is as follows: In the V-cycle an approximation to the exact solution is relaxed on and then transferred to a coarser grid. The process is repeated recursively until the coarsest grid is reached where the corresponding residual equation is inexpensive to solve. The approximation is then transferred back to finer grids until the finest grid is reached. In this way all error components can be reduced.

In geometric multigrid, the grid with the greatest number of points (the fine grid) is called Ω^h, the next coarser grid is Ω^{2h}, the third coarser grid is Ω^{4h}, and so on. Successive grids are often selected by halving the number of points in each

dimension. Therefore, Ω^h has 2^d as many points as Ω^{2h}, where d is the dimension of the problem.

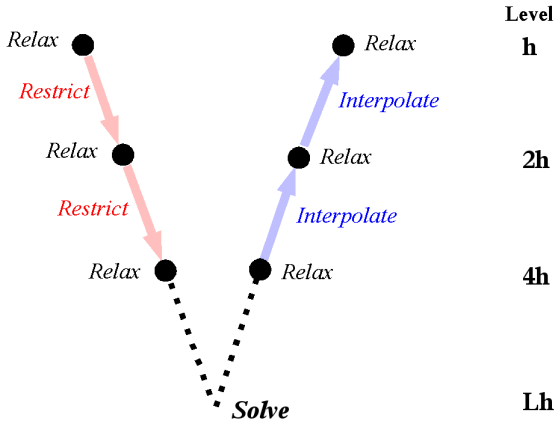


Figure 2: Grid configuration for a V-cycle.

3.1 The multigrid algorithm

For simplification, I will describe the multigrid algorithm with the following two grid correction schemes. After a few smoothing sweeps we obtain an approximation v^h whose error $e^h = u^h - v^h$ is smooth. Then we need to express this smooth error as a solution of a coarse problem, whose matrix A^{2h} and right hand side f^{2h} have to be defined. The residual $r^h = f^h - A^h v^h$ is a smooth function if e^h is smooth. Obviously, the original equation $A^h u^h = f^h$ and the residual equation $A^h e^h = r^h$ are equivalent. The difference is that $e^h$and r^h are smooth, so we can represent them on a coarser grid, Ω^{2h}, with half the number of points on the fine-grid. Define r^{2h} as the restriction of the fine-grid residual to the coarse-grid, $r^{2h} = I_h^{2h} r^h$, where I_h^{2h} is a restriction operator. In this way the right-hand side of the coarse-grid problem is defined. The coarse problem is $A^{2h} e^{2h} = r^{2h}$. After a few smoothing sweeps, the error e^h is smooth, so we can apply an interpolation operator, I_{2h}^h, to transfer e^h to the fine-grid. Then, we update v^h by applying the following coarse-grid correction step $v_{new}^h = v^h + I_{2h}^h e^{2h}$. In practice, the interpolation procedure may also introduce high-frequency error components on the fine-grid. Therefore it is convenient to complete the procedure by applying ν_2 post-smoothing sweeps after the coarse-grid correction. Since the coarse-grid problem is not much different from the original problem, we can apply the two-grid correction scheme to the residual equation on Ω^{2h}, relaxing there and then moving to Ω^{4h} for the correction step. This process can be repeated on coarser grids until a direct solution of the residual equation is reached.

Assuming that successive coarser grids have been defined, and assuming that a matrix operator A is defined on all grids, the multigrid algorithm, by using V-cycle, can be summarized as follows [54]:

$$v^h \leftarrow MG\left(v^h, f^h\right)$$

while convergence criteria are not satisfied , perform a V-cycle:

- Relax on $A^h u^h = f^h$ ν_1 times with initial guess v^h.
- Compute the residual $r^h = f^h - A^h u^h$ and restrict it to the coarse-grid by $f^{2h} = I_h^{2h} r^h$.
  - Relax on $A^{2h} u^{2h} = f^{2h}$ ν_1 times with initial guess $v^{2h} = 0$.
  - Compute $f^{4h} = I_{2h}^{4h} r^{2h}$.
    - Relax on $A^{4h} u^{4h} = f^{4h}$ ν_1 times with initial guess $v^{4h} = 0$.
    - Compute $f^{8h} = I_{4h}^{8h} r^{4h}$.

      .
      .
      .

      · Solve $A^{Lh} u^{Lh} = f^{Lh}$.

      .
      .
      .
    - Correct $v^{4h} \rightarrow v^{4h} + I_{8h}^{4h} v^{8h}$.
    - Relax on $A^{4h} u^{4h} = f^{4h}$ ν_2 times with initial guess v^{4h}.
  - Correct $v^{2h} = v^{2h} + I_{4h}^{2h} v^{4h}$.
  - Relax on $A^{2h} u^{2h} = f^{2h}$ ν_2 times with initial guess v^{2h}.
- Correct $v^h = v^h + I_{2h}^h v^{2h}$.
- Relax on $A^h u^h = f^h$ ν_2 times with initial guess v^h.

Here L is used to label the coarsest grid. The integers ν_1 and ν_2 are parameters in the scheme that control the number of relaxation sweeps before and after visiting the coarse-grid. They are usually fixed at the start based on past experimental results. The multigrid work-flow is pictured in Figure 3.

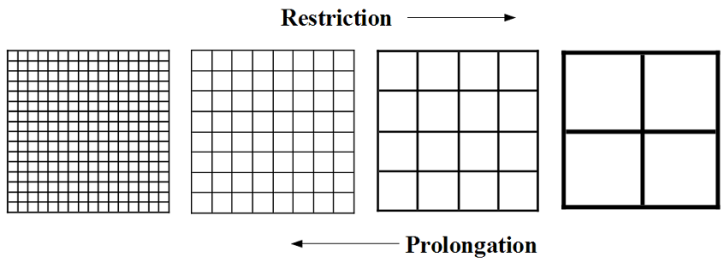


Figure 3: Geometric multigrid setting.

3.2 The Full Multigrid (FMG) method

It is always beneficial to start the iterative procedure from a good initial approximation. The multigrid method suggests a natural way of getting this approximation cheaply. Starting the solution process from a coarse-level, for example Ω^{2h}, where the discretized problem $A^{2h}u^{2h} = f^{2h}$ is easily solved. The solution can be interpolated to the next finer level as an initial approximation for the iterative process to solve $A^h u^h = f^h$. The interpolation I_{2h}^h is applied and then the multigrid solution process at level Ω^h is started. This idea of using a coarse-level approximation as a first guess on a finer level is known as nested iteration. The algorithm that joins nested iterations with the V-cycle is called the full multigrid V-cycle. Given first an explicit term, the FMG algorithm for two levels is written as follows:

Initialize $f^{2h} \leftarrow I_h^{2h} f^h$,

- solve or relax on coarsest level, $A^{2h}u^{2h} = f^{2h}$,
- $v^h \leftarrow I_{2h}^h v^{2h}$.
- Relax $A^h u^h = f^h$ ν_0 times.

The coarse-level right sides are initialized by transferring f^h from the fine-level. The cycling parameter ν_0 sets the number of V-cycles applied at each level. Practical experience shows that it is problem dependent. For example, in subsonic flow, where the equation is elliptic, $\nu_0 = 1$ is a a good choice. The extension to multilevel is straightforward. The schedule of grids for FMG with $\nu_0 = 1$ is shown in Figure 4.

The FMG scheme costs a little more per cycle than the V-cycle scheme (see for example [54]). However, a properly designed FMG can be much more effective because before the Ω^h problem is even touched, the Ω^{2h} problem has already been solved to the level of discretization error. This is because the iterations on the coarse-level are designed to provide a good initial guess for the next finer level. In this work an FMG (containing two levels only) is applied as a part of the AMG code for the nonlinear problems involving circulation. See the details in the following sections.

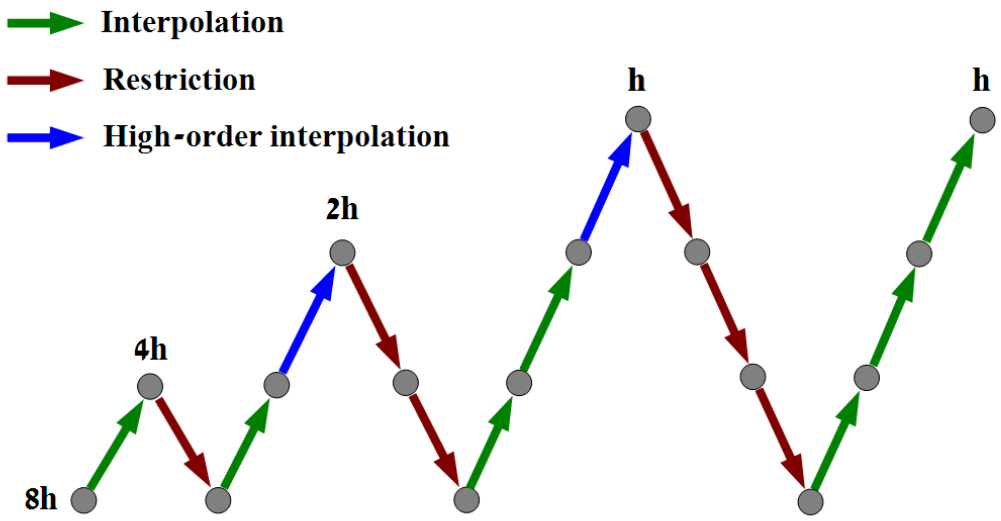


Figure 4: Schedule of grids for the FMG scheme applied with four levels.

3.3 Measuring the algorithm's efficiency

An important question is, how much do the multigrid schemes cost in terms of storage and computation time?

Storage Cost - A d-dimensional grid with n points in each dimension has n^d points. Two arrays must be stored, v and f, in the finest level for $2n^d$ total storage locations. Assuming that Ω^{2h} is recursively constructed by halving the number of points in each dimension of Ω^h, the coarse-grids require 2^{-d} times the amount of storage of the next finest grid. Therefore, the total storage requirement can be expressed as follows:

$$Storage\ cost\ = 2n^d \left\{1 + 2^{-d} + 2^{-2d} + ... + 2^{-nd}\right\} < \frac{2n^d}{1 - 2^{-d}}. \qquad (1)$$

Therefore, in one, two, and three dimensions, the total storage cost is less than 2, $4/3$, and $8/7$ times the cost of storage of the fine-grid quantities, respectively.

Computational cost - It is convenient to measure the computational cost of multigrid schemes in terms of work units (WU), which is the cost of performing one relaxation sweep on the finest grid. Since the coarse-grid consists of 2^{-d} times the number of points as on the next finest grid, the work required on a coarse-grid is 2^{-d} times the amount of work required on the next finest grid. Each level is visited twice , and assuming that a $v(1,1)$ cycle is employed, then the computational cost is computed as follows:

$$Computational\ cost = 2\left\{1 + 2^{-d} + 2^{-2d} + ... + 2^{-nd}\right\} WU < \frac{2}{1 - 2^{-d}} WU. \qquad (2)$$

In one, two, and three dimensions, this corresponds to 4, 8/3, and 16/3, respectively. As one can see, the computational cost, measured in terms of WU, is not affected by the problem size. Since this result is desirable for the purpose of scalability, it does not indicate how multigrid performs compared to other algorithms in terms of total computational cost. In order to get this comparison, the number of V-cycles, which is required to reduce the error, needs to be known. This parameter is calculated below.

There are two forms of errors present in the numerical solution of a PDE problem. The first is called the discretization error and is a direct result of moving from the continuous problem u to the discrete problem v^h. In a 1D problem, the discretization error is defined by

$$E_i^h = u(x_i) - u_i^h, \quad 1 \leq i < n-1. \tag{3}$$

Using the definition of the discrete L_2 norm, it can be shown that (3) can be bounded by

$$\left\|E^h\right\|_h \leq Kh^p \tag{4}$$

where K is a positive constant and p is a positive integer (1 for a 1D problem and 2 for a 2D problem).

A second type of error is generated while approximating the exact solution of the discrete problem u^h with v^h. This is the algebraic error and is simply defined as

$$e^h = u^h - v^h. \tag{5}$$

The goal of an iterative method is to reduce the algebraic error as fast as possible and attain a result that is of the order of the discretization error. Consider a d-dimensional problem with n^d unknowns such that the grid spacing is $h = \frac{1}{n}$ in each dimension. Assume that the convergence factor γ is bounded in each V-cycle (independent on h). If the scheme reduces the algebraic error from $O(1)$ to the level of the discretization error, then, $O(h^p) = O(n^{-p})$, the number of V-cycles required, ν, must satisfy $\gamma^\nu = O(n^{-p})$ and hence $\nu = O(logn)$. The cost of one V-cycle is $O\left(n^d\right)$. In order to get an error that is in the level of discretization while using the multigrid algorithm, the cost in this case is $O\left(n^d logn\right)$. This result is much better than the direct methods, for example Gauss elimination, where $O\left(n^{3d}\right)$ is obtained.

4 The Basic AMG Method

We address the reader to [69, 54, 75, 70] for a detailed description of the AMG algorithm, while in this section we shall only briefly review the algorithm emphasizing its aspects that require special treatment for the purpose of this work.

Consider a certain boundary-value problem for a scalar PDE in domain Ω^m. Its discretization will result in a linear algebra problem of the form $A^m u^m = f^m$, where A^m is an $n \times n$ matrix with entries a_{ij} with $i = 1, ..., n$, $j = 1, ..., n$, $m = 1, 2, ...$, $u^m = \{u_j\}$ is the vector of unknowns, $f^m = \{f_j\}$ is the forcing term vector, and n is the number of points in the computational grid covering the domain. These equations formally play the same role as the coarse-grid equations defined in the geometric multigrid method. A grid Ω^m can be regarded as a set of unknowns u_j^m.

In any multigrid method there are two basic processes involved: smoothing and coarse-grid correction. Their purpose is to reduce all error components in the overall iteration. In geometric multigrid methods the coarse-grids are uniformly coarsened or semi-coarsened; thus the freedom in the selection of the coarse-grids is limited. The grid's hierarchy is constructed based on the grid geometry information rather than properties of the matrix A. In algebraic multigrid we have no access to the geometry of the problem. The construction of the coarse-levels and inter-grid transfer operators are based on the properties of the discrete operator only.

In a geometric multigrid the definition of smoothness of the error involves grid geometry. The absence of grids in AMG renders this definition meaningless. Therefore, the concept of smoothness has to be generalized to some meaningful measurable quantity that can be computed based on the discrete operator only. A common definition of the algebraic smoothness is based on the fact that a simple pointwise relaxation scheme, like symmetric Gauss–Siedel (SGS), effectively damps highly oscillatory modes of the error only. Consequently, the coarse-grid correction must deal with the remaining slow components. The characterization of such slow components ,e, is: $Ae \simeq 0$.

The AMG algorithm includes two parts. The first part is an automatic setup phase that performs four steps:

1. The matrix entries are analyzed. The purpose is to find strong and weak connections between the variables.

2. Construction of the next coarser level based on a certain splitting criterion. All the variables are split to C-points (those are going to form the next coarse-level) and F-points.

3. The transfer operators, restriction and interpolation (I_{m+1}^{m}, I_{m}^{m+1}), are evaluated. Usually, the restriction is the transpose of the interpolation.

4. The coarse-level matrix is computed, usually based on the Galerkin principle $A^m = I_{m+1}^{m} A^{m+1} I_{m}^{m+1}$. In this paper another algorithm is presented for the construction of the restriction operator and the coarse-level matrix.

The second part, the solution phase, uses the resulting operators in order to perform multigrid cycles until a desired level of accuracy is reached. This phase consists of three components:

1. Smoothing by a relaxation method such as Jacobi or Gauss–Seidel.

2. Transfer data between levels by restriction and interpolation.

3. Solution on the coarse-level.

We shall give here a brief description of these phases of what is considered a classical AMG in the spirit of [69] followed by a description of the solution process. Only the aspects of the processes that require a special attention for the purpose of this work will be described in more detail.

4.1 Strong and weak connections

In the context of an AMG we are going to deal at each level with a linear system of equations

$$A^m u^m = f^m \tag{6}$$

where m is the level index. The goal is to split variables into two groups: those remaining on the coarse-level (C) and those that can be "represented" by the coarse-points and, therefore, may not be included in the coarse-level (belong to $F \backslash C$). As mentioned before, the coarsening process is derived based on the strong and weak connections between unknowns, which essentially measure the relative size of the off-diagonal entries. Connections between neighboring variables are considered strong if the size of the corresponding matrix entry exceeds a certain threshold, relative to the maximum entry of the row. This threshold value is very important for constructing a good coarse-grid. According to [69], a point i is said to be strongly connected to a point j, if

$$-a_{ij}^m \geq \varepsilon \max_{k \neq i} \left(-a_{ik}^m\right). \tag{7}$$

The threshold value ε is kept fixed for most applications, with a typical value of 0.25. It was found in this work, however, that for the problems considered here this approach can lead to an inadequate coarsening and interpolation processes, and hence to poor convergence. Therefore, a dynamic threshold approach was devised (see Section 7).

Before proceeding with the coarsening process we need to make one more definition. Denote S_i^m as the set of points that strongly influence the point i; that is, the point upon which the point i strongly depends. Also, denote by $\left(S_i^m\right)^T$ the points that strongly depend on the point i. The strength matrix $S_{i,j}^m$ is defined as:

$$S_{i,j}^m = \begin{cases} 1 & \text{if } i \neq j \text{ and } u_i \text{ strongly depends on } u_j \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The nonzero entries in row i of S_i^m, and the nonzero entries in column i of S_i^m indicate the points in $\left(S_i^m\right)^T$. Now the interpolation process can be introduced.

4.2 The standard interpolation operator

Let us define the set of points that are strongly connected to i by S_i^m. Let $N_i^m = \left\{j \in \Omega^m : j \neq i,\ a_{ij}^m \neq 0\right\}$ denote the neighborhood of a point $i \in \Omega^m$. $D_i = N_i^m - C_i^m$, $D_i^s = D_i \cap S_i$, $D_i^w = D_i - D_i^s$.

The neighboring coarse-grid points are denoted by C_i^m. The neighboring fine-grid points denoted by D_i^s, and the point that is weakly connected to the point i are denoted by D_i^w. Denote the neighboring fine-grid strongly connected to point i by D_i^s and the fine-grid points weakly connected to i by D_i^w.

The interpolation of the error at the F-point i takes the form

$$e_i^m = \sum_{j \in C_i^m} w_{ij}^m e_j^{m+1} \qquad \forall i \in \Omega^m, \tag{9}$$

where w_{ij}^m is an interpolation weight, and C_i^m is the subset of C-points whose values will be used to interpolate a value at i.

Since the error e_i^m is obtained after a smoothing process, then the ith equation becomes

$$a_{ii}^m e_i^m + \sum_{k \in C_i^m} a_{ik}^m e_k^m + \sum_{j \in D_i^s} a_{ij}^m e_j^m + \sum_{j \in D_i^w} a_{ij}^m e_j^m \approx 0 \qquad \forall i \in \Omega^m. \tag{10}$$

In order to determine the weights, w_{ij}^m, we need to replace e_j^m in the second and third sums with approximations in terms of e_k^m, where $k \in C_i^m$. Thus, replacing e_j^m with e_k^m corresponds to taking into account strong $F-F$ connections using C-points that are common between the F-points. Since the error is smooth, the error introduced in making this approximation is insignificant. By applying this distribution we get the following interpolation formula for the variable $i \in F^m$ as presented in [69],

$$w_{ij}^m = -\frac{a_{ij}^m + \sum_{k \in D_i^s} \frac{a_{ik}^m a_{ki}^m}{\sum_{l \in C_i^m} a_{kl}^m}}{a_{ii}^m + \sum_{l \in C_i^m} a_{il}^m}. \tag{11}$$

Experience has shown that this proposed interpolation produces accurate results, and it is efficient mainly for the symmetric and M-matrices, but when it is applied to a matrix with both positive and negative off-diagonal entries, especially when the negative off-diagonal values are large (as, for example, in the case of transonic flow), the AMG method could fail during the setup phase. The details are discussed in Section 7.

4.3 Classical Ruge and Stueben coarsening

The classical Ruge and Stueben coarsening algorithm will be presented here. For further details the reader is referred to [55, 69]. Once strong connections are determined (see (8) for definition of strong dependence and influence), a coarse-grid is chosen so that all strongly connected neighbors of any fine-grid point are available for direct or indirect interpolation. The Ruge and Stueben algorithm selects coarse-grid points based on two heuristic criteria:

- **H-1**: For each fine-grid point i, each point $j \in S_i$ must either be a coarse-grid neighbor or strongly depend on at least one coarse-grid neighbor of i in S_i.

This criterion tends to create large coarse-grids. In practice, a second heuristic is used to limit the size of the coarse-grid.

- **H-2**: The set of coarse-grid points should be a maximal subset of all F-points with the property that no C-point strongly depends on another C-point.

The motivation for **H-1** follows from the fact that an effective coarsening scheme should allow accurate interpolation. Since smooth error varies slowly in the direction of strong connection, smooth error will be interpolated well by points that are strongly connected.

The heuristic **H-2** is much easier to satisfy, and is in place to ensure that coarse-grids are sufficiently coarsened in order to keep computational cost at scalable levels. Since these two rules may be contradictory, a typical AMG algorithm

relays on enforcing **H-1**, using **H-2** as a guide. The reduction in computational cost per V-cycle that can be achieved by enforcing **H-2** rigorously is lost in the convergence behavior of the iteration. Because **H-2** is weaker condition than **H-1**, the selection of coarse-grid points is accomplished by a two-pass process that picks a set satisfying **H-2** (initial partition of the grid to C and F-points), then checks for any points where **H-1** is violated, adding new C-points if this occurs. In the first pass the C-points are selected based on their number of strongly connected neighbors. A suitability measure of a point i to be a C-point is simply to count the number of points strongly influenced by i. This count is λ_i and is simply the column sum of column i of the strength matrix S. The greater the value of λ_i , the more useful point i will be in interpolation if defined as a C-point. The point with the largest weight is then selected as a C-point. Each $j \in S_i^T$ is now strongly connected to a C-point; all these j points are assigned as F-points so that **H-2** is not violated. All strongly connected neighbors of these j points are then more attractive to be a C-point, thus their weights is increased. In this way the points that strongly influence j are more likely to be chosen as C-points. This process of choosing C-points based on maximal measure, and making all strongly influenced points F-points, is then repeated. This process is continued until all points are either C- or F-points. An example of the first pass is illustrated for a structured grid in Figure 3. Edges indicate strong connections, a black point indicates C-points, and a white dot indicates F-points. A white dot with a number indicates an unassigned point, while the number corresponds to its measure in the corresponding stage in the coarsening process. After a point is assigned, all edges connected to that point are removed.

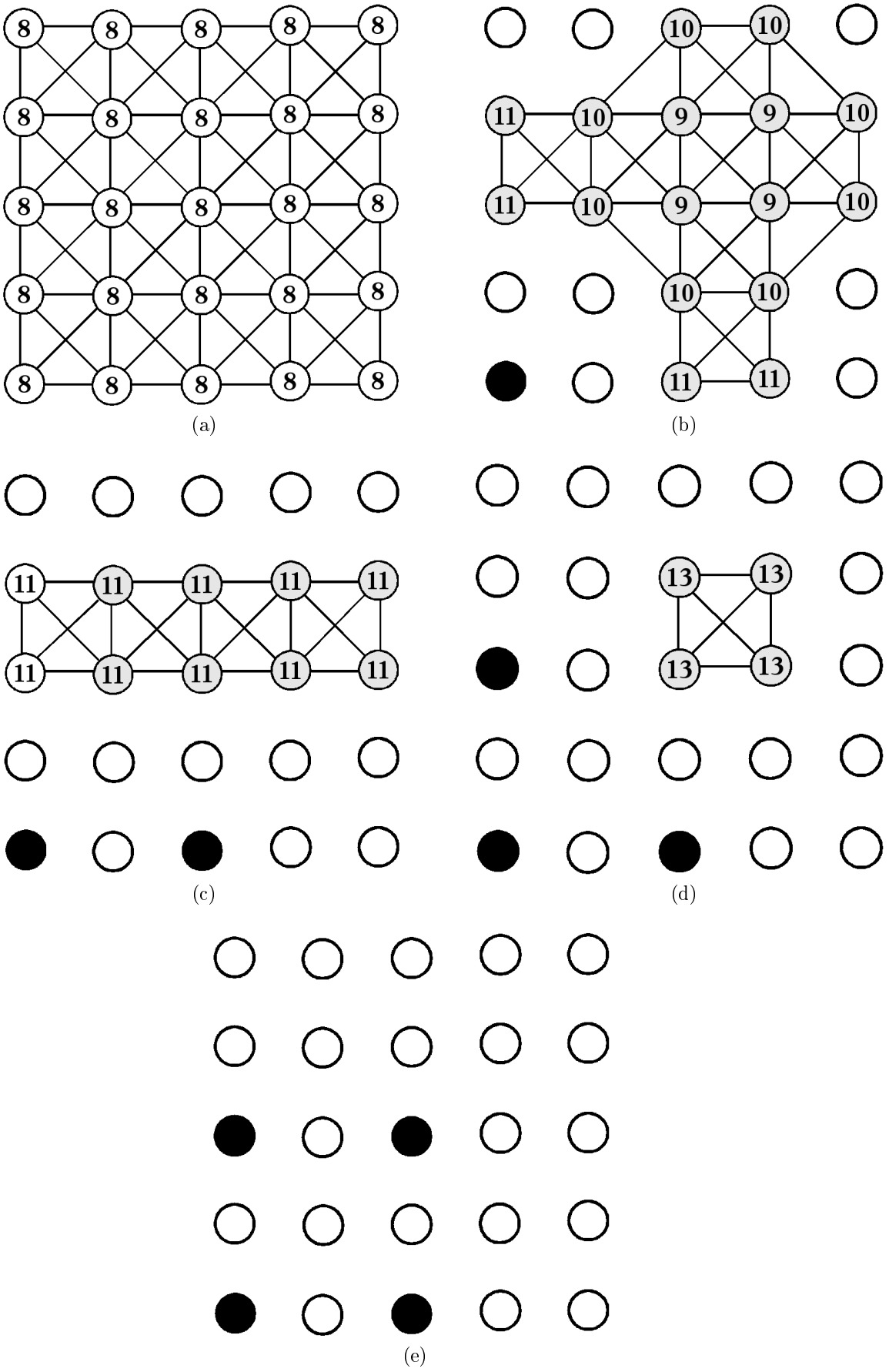


Figure 5: An illustration of the first pass of the algebraic multigrid coarsening algorithm for a nine-point discretization stencil on a uniform grid with periodic boundary conditions. The upper left (a) diagram is the original grid, the lower center (e) is the final step.

The algorithm in Figure 5 is summarized as follows:

1. The nodes of the graph of the strength matrix are assigned a weight according to the number of off-diagonal connections.
2. A point with the maximal weight is chosen as a C-point.
3. The neighbors of the new C-point are set to be F-points.
4. For each new F-point, the weights of its neighbors are increased by one to make them more likely to be chosen next. This process is repeated until all points are either C- or F-points.

In this way an initial coarse-grid is chosen that gives a maximal independent set over all strong connections. The purpose of the second pass is to enforce **H-1** rigorously. Additional points are then added to the coarse-grid, if necessary. This means that some F-points are changed to C-points. This is done so that a minimum number of C-points are added. In the case of the example in Figure 5(E), the resulting grid becomes that shown in Figure 6.

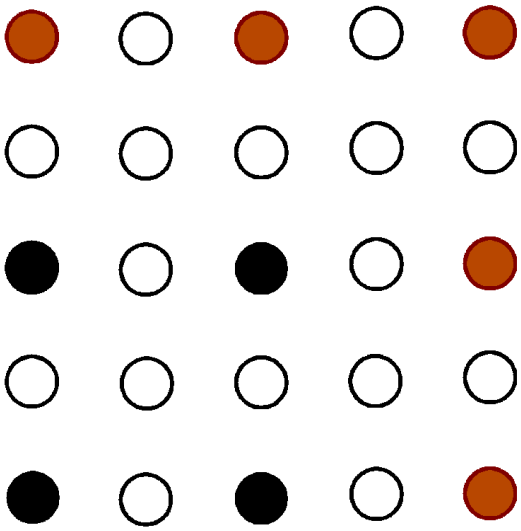


Figure 6: Second pass process for the nine-point Laplacian problem with periodic boundary conditions. The added C-points are shown in the final coloring as red dots with heavy red outlines.

Aggressive coarsening

A very efficient way of improving memory is by aggressive coarsening instead of the standard coarsening. In this approach strong connectivity is also defined by strongly connected F-points, which means that F-to-C connectivity does not need direct connections to the C-point.

Aggressive coarsening is implemented by first extending the definition of strong connectivity to variables that are not connected directly through a given stencil. In practice, each F-point is surrounded by C-points from which it is interpolated. Instead of using only the C-points in the stencil we take into account all the strong connections of each C-point in the stencil. It is usually sufficient to apply aggressive coarsening only in the first coarsening step and maintain standard coarsening for the rest of the coarse-levels. This coarsening approach results in significantly improved complexities and memory requirements, but the convergence speed is reduced since fewer points on the coarse-level results in less efficient interpolation. For further details about the aggressive coarsening approach the reader is referred to Stueben [69].

4.3.1 Restriction and prolongation (interpolation) operators

Having constructed the coarse-levels, we need to devise the restriction (residual transfer) and prolongation (correction interpolation operators). According to the classical AMG approach (see [69] for details), one can design the prolongation operator I_m^{m+1}, while the restriction operator is taken to be the transposed of the prolongation

$$I_{m+1}^m = \left(I_m^{m+1}\right)^T. \tag{12}$$

For the purpose of this work, however, this procedure had to be modified so that the prolongation operator is devised directly according to certain rules (see Section 7).

4.3.2 Restriction and coarse-level operators

As mentioned, in the standard approach suggested by Stueben [69], the restriction operator is defined as the transpose of the interpolation I_{m+1}^m, I_m^{m+1}. Then the coarse-grid operator is defined by the Galerkin type procedure,

$$A^{m+1} = I_m^{m+1} A^m I_{m+1}^m. \tag{13}$$

This is the simplest way to construct the restriction and coarse-grid operators. However, it is known (see, for instance, [76]) that it may lead to poor convergence when matrix A^m is not an M-matrix, which is normally the case for the problems considered in this work. Therefore, some alternatives had to be considered (see Section 7).

4.3.3 Solution phase

The solution phase of the algorithm relies on the resulting operators to perform an iterative solution process (AMG cycles) until a desired level of accuracy is reached. An AMG cycle can be described in the following as consisting of the following stages:

1. Pre-smoothing, by a relaxation method like Jacobi or Gauss–Seidel
2. Performing restriction of the problem, i.e., transferring residuals to the coarser level.

3. Solution on the coarse-level (by recursion).

4. Performing prolongation, i.e., interpolating the solution correction to the finer level.

5. Post-smoothing, again, by a certain relaxation method.

Usually, the relaxation used as an ingredient of an AMG algorithm is a pointwise one. One of the central contributions of this work is developing a stable pointwise direction-independent relaxation for the entire range of the flow speed, from low Mach number flow to transonic and supersonic regions (see Section 6). This development was a prerequisite for considering an application of AMG to the transonic flow problem.

4.3.4 Measuring complexity

The computational complexity concept is intended to measure the algorithm's requirements for computer resources:

- computer storage (memory),
- CPU time.

There are four types of complexity measure that are commonly considered: convergence rate, grid complexity, operator complexity, and stencil size.

Convergence rate

It is defined as the average fractional change in the residual for a multigrid cycle. Foe example, a convergence factor of 0.1 indicates that the residual decreases by one order of magnitude in each multigrid cycle on average over all computed cycles.

Grid complexity

It is defined as the total number of elements in the coarse-levels divided by the number of elements in the fine-level. Let n_Ω^k denote the number of degrees of freedom on level k and n_L^k the number of nonzero entries in the level k operator A^k . Therefore the grid complexity is computed as follows:

$$C_\Omega = \frac{\sum_{k=1}^{m} n_\Omega^k}{n_\Omega^1}. \tag{14}$$

Grid complexity provides a direct measure of the storage required for the solution and right side vectors, and it is a useful tool to compare different coarsening strategies. In geometric multigrid, if coarse-grids are constructed by halving the number of points in each dimension, the grid complexities for one, two, and three dimensions are 2, 4/3, and 8/7, respectively [54].

Operator complexity

It is defined as the sum of the nonzero matrix elements in all the coarse-levels, divided by the number of the nonzero matrix elements on the fine-level. The operator complexity is defined by

$$C_L = \frac{\sum_{k=1}^{m} n_L^k}{n_L^1}. \tag{15}$$

The amount of work required by the relaxation and residual computations is directly proportional to the number of nonzeros on the coarse-levels. Therefore, low value of the operator complexity that increase linearly with the problem's resolution signifies a linear complexity operator.

Stencil size

The stencil size is the average number of points in a matrix row. It strongly influences the setup time (the time to compute the coarse-levels and operators), since growing stencil sizes substantially increases the number of operations in the coarsening and interpolation processes. It also influences the relaxation sweep time. In order to obtain a scalable algorithm, both operator complexity and stencil size should be independent of the problem size.

Before we proceed, however, it is useful to give a brief summary of the two main relaxation methods that usually serve as smoothers for both algebraic and geometric multigrid approaches.

5 Relaxation methods

The area of algebraic systems has been treated extensively in the literature, with the aim of improving the algorithms and reducing the number of operations. Some direct and iterative methods can be found in [77, 78, 79, 80, 81, 82, 83]. Relaxation methods are represented in this work by the Jacobi and Gauss–Seidel iterations applied to linear system of equations. These classical relaxation methods are robust and relatively easy to implement. They are applicable in more general cases than most direct methods [79, 84]. However, they are known to be rather slow to converge.

5.1 Point Jacobi method

Let us consider the algebraic system $Ax = b$, where A is an $n \times n$ matrix. The ith row of the equation is $\sum_j a_{i,j}x_i = b_i$. This can be written as:

$$\begin{aligned} \sum_{j<i} a_{i,j}x_j + a_{ii}x_i + \sum_{j>i} a_{ij}x_i &= b_i \\ a_{ii}x_i = b_i - \sum_{j<i} a_{i,j}x_j + \sum_{j>i} a_{ij}x_i & \\ x_i = \frac{\left(b_i - \sum_{j<i} a_{ij}x_j + \sum_{j>i} a_{ij}x_i\right)}{a_{ii}}. & \end{aligned} \tag{16}$$

This suggests the following iterative process,

$$x_i^{(k+1)} = \left(\frac{\left(b_i - \sum_{j<i} a_{ij} x_j^{(k)} + \sum_{j>i} a_{ij} x_i^{(k)} \right)}{a_{ii}} \right), \tag{17}$$

where k is an iteration index. The general formulation of the Jacobi method is best represented in matrix form. We decompose the matrix A in a sum of three matrices containing the main diagonal D, the upper triangular part U, and the lower triangular L. That is, we write $A = D - L - U$. Then the linear system $Au = f$ becomes

$$\begin{aligned} (D - L - U)\, u &= f \\ Du &= (L + U)\, u + f. \end{aligned} \tag{18}$$

Multiplying by D^{-1} corresponds exactly to solving the jth equation for x_j for all $j = 1, ..., n$,

$$u = D^{-1} (L + U)\, u + D^{-1} f. \tag{19}$$

Let define the Jacobi iteration matrix by $R_j = D^{-1} (L + U)$, then the iteration process of the Jacobi method is $v^{(1)} = R_j v^{(0)} + D^{-1} f$, while the current approximation (or the initial guess) on the first iteration is denoted $v^{(0)}$, while the new, updated approximation is denoted $v^{(1)}$. In practice, once all of the $v^{(1)}$ have been computed, the procedure is repeated until convergence to the solution is obtained. An important modification that can be made to the Jacobi method is by a weighted average:

$$v^{(1)} = [(1 - w)\, I + w R_j]\, v^{(0)} + w D^{-1} f. \tag{20}$$

This new iteration is called the damped Jacobi method. When $w = 1$ we get the original Jacobi iteration. The Jacobi iteration matrix is defined by $R_w = (1 - w)\, I + w R_j$; then the method can be expressed as

$$v^{(1)} = R_w v^{(0)} + w D^{-1} f. \tag{21}$$

5.2 Gauss–Seidel method

The Jacobi method does not use all the available up-to-date information when updating $x_i^{(k+1)}$. It uses values from the k^{th} iteration for all x_j, even for $j < i$ where $x_j^{(k+1)}$is already known. If we revise the Jacobi method so that we always use the most current estimate of the exact x_i, then we obtain the Gauss–Seidel iteration:

$$x_i^{(k+1)} = \left(\frac{\left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} + \sum_{j>i} a_{ij} x_i^{(k)} \right)}{a_{ii}} \right). \tag{22}$$

In the matrix form, splitting the matrix A into the form $A = D - L - U$, we can write the system of equations as

$$u = (D - L)^{-1} U u + (D - L)^{-1} f. \tag{23}$$

We can define the Gauss–Seidel iteration matrix by

$$R_G = (D - L)^{-1} U, \tag{24}$$

and the method is

$$v^{(1)} = R_G v^{(0)} + (D - L)^{-1} f. \tag{25}$$

For the Jacobi method, the order in which the components of v are updated is immaterial, since components are never overwritten. For Gauss–Seidel, the order of updating is significant. Instead of sweeping through the grid points in ascending order, we can sweep over the grid points in descending order. We can also alternate between ascending and descending order. This method is called the symmetric Gauss–Seidel method. One iteration of this method consists of two consecutive relaxations: a standard Gauss–Seidel relaxation, followed by another Gauss–Seidel relaxation in the reverse order.

5.3 Convergence analysis

The update formula, for both Jacobi and Gauss–Seidel methods, is linear in v and does not change from one relaxation to the next. This type of formula is known as stationary linear iteration. Each of the methods described above may be represented in the form:

$$v^{(k+1)} = Rv^k + g, \tag{26}$$

where R is a general iteration matrix. These methods, obviously, when applied to the exact solution, should not change it. Let us verify this for the Jacobi method

$$\begin{aligned} v^{(k+1)} &= R_J v^{(k)} + D^{-1} f, \\ v^{(k+1)} &= R_J v^{(k)} + D^{-1} A u, \\ v^{(k+1)} &= D^{-1} (L + U) v^{(k)} + D^{-1} (D - L - U) u, \\ v^{(k+1)} &= D^{-1} (L + U) v^{(k)} - D^{-1} (L + U) u + u, \end{aligned} \tag{27}$$

and thus

$$v^{(k+1)} = v^{(k)} = u \text{ if } v^{(k)} = u. \tag{28}$$

In general it can be written as follows:

$$u = Ru + g. \tag{29}$$

By subtracting (27) from (29) we find that

$$u - v^{(k+1)} = Ru + g - \left(Rv^{(k)} + g\right) = R\left(u - v^{(k)}\right), \tag{30}$$

and thus

$$e^{(k+1)} = Re^{(k)}. \tag{31}$$

If the iteration is performed m times we get

$$e^{(m)} = R^m e^{(0)}. \tag{32}$$

The superscript 0 corresponds to the initial approximation. It is possible to bound the error after m iterations by

$$\left\|e^{(m)}\right\| \leq \|R^m\| \left\|e^{(0)}\right\|. \tag{33}$$

It follows that the error is forced to zero as the iteration proceeds, if and only if $\|R^m\| < 1$. The norm of the error will approach zero as the number of iterations increases, only if $\rho(R) < 1$. The spectral radius $\rho(R)$ is also called the *asymptotic convergence factor.*

6 Transonic flow – model problem formulation

6.1 The continuity equation

The law of mass conservation expresses the fact that mass cannot be created in a fluid system and it cannot disappear. Consider a finite control volume fixed in space and time Ω, as sketched in Figure 7. The time rate of change of the total mass inside the control volume Ω is

$$\frac{\partial}{\partial t} \int_{\Omega} (\rho d\Omega). \tag{34}$$

The mass flow of a fluid through a surface element ds fixed in space is $\rho \overrightarrow{V} \cdot \overrightarrow{n} ds$, where $\hat{n}$ is a unit normal to the surface. The convection for the case of closed domain is that $\overrightarrow{n}$ always points outwards. If the following product is negative $\overrightarrow{V} \cdot \overrightarrow{n} < 0$, it is called an inflow boundary (mass enters the domain), and if this product is positive it is called an outflow boundary (mass leaves the domain). By taking into account the general formulation of (34), the conservation law of mass is

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial\Omega} \rho \left(\overrightarrow{V} \cdot \overrightarrow{n}\right) dS = 0. \tag{35}$$

The latter result is known as the Gauss theorem [12].

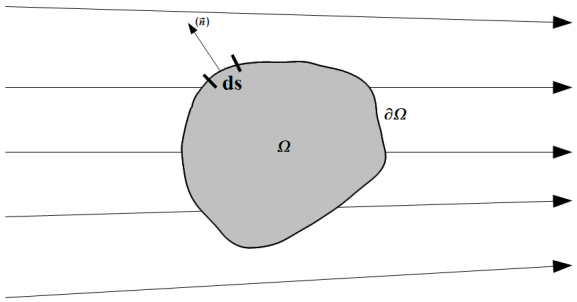


Figure 7: Definition of a finite control volume.

6.2 The momentum equation

The variation of momentum is caused by the net force acting on the mass element. The momentum of a mass element with a control volume Ω is

$$\rho \overrightarrow{V} d\Omega. \tag{36}$$

The variation in time of the momentum is

$$\frac{d}{dt} \int \left(\rho \overrightarrow{V} d\Omega \right). \tag{37}$$

The transfer of momentum across the boundary of the control volume is described by a convective flux tensor that consists of the following two components: $\rho u \overrightarrow{V}$ and $\rho v \overrightarrow{V}$. The contribution of the convective flux tensor to the conservation of momentum is then

$$- \oint_{\partial\Omega} \rho \overrightarrow{V} \left(\overrightarrow{V} \overrightarrow{n} \right) dS. \tag{38}$$

Since there is no diffusion of momentum when the fluid is at rest, the diffusive flux is zero. There are two kind of forces acting on the control volume:

• Body forces - acts directly on the mass element, i.e., gravitational buoyancy, Coriolis or centrifugal forces.

• Surface forces - acts directly on the surface of the mass element. These forces result from the pressure distribution imposed by the outside fluid surrounding the volume, and shear and normal stresses resulting from the friction between the fluid and the surface of the volume.

The contribution of the body force f_e to the momentum conservation is $\int_{\partial\Omega} \rho \overrightarrow{f_e} d\Omega$. The surface sources consist of the pressure component and of a viscous stress tensor τ $Q_s = -p\overline{\overline{I}} + \overline{\overline{\tau}}$. Summing the above contributions to the general conservation law, we finally obtain the conservation of momentum:

$$\frac{\partial}{\partial t}\int\limits_{\Omega}\rho\overrightarrow{V}d\Omega+\oint\limits_{\partial\Omega}\rho\overrightarrow{V}\left(\overrightarrow{V}\overrightarrow{n}\right)dS=\int\limits_{\Omega}\rho\overrightarrow{f_e}d\Omega-\oint\limits_{\partial\Omega}p\overrightarrow{n}dS+\oint\limits_{\partial\Omega}\left(\overline{\overline{\tau}}\cdot\overrightarrow{n}\right)dS. \quad (39)$$

6.3 The energy equation

The First Law of Thermodynamic states that any changes in time of the total energy are caused by the rate of work of forces acting on the volume and the net heat flux into it. The total energy per unit mass E is obtained by adding the internal energy per unit mass, e, to the kinetic energy per unit mass $V^2/2$,

$$E=e+\frac{V^2}{2}. \quad (40)$$

The total energy per unit volume is ρE and its variation in time within the volume Ω is expressed as $d/dt\int_\Omega\rho E d\Omega$. Then, the contribution of the convective flux is

$$-\oint\limits_{\partial\Omega}\rho E\left(\overrightarrow{V}\cdot\overrightarrow{n}\right)dS. \quad (41)$$

The diffusive flux F_D is proportional to the gradient of the conserved quantity per unit mass (Fick's law),

$$\vec{F}_D=-\gamma\rho\kappa\nabla e, \quad (42)$$

where $\gamma=C_p/C_v$ is the ratio of specific heat coefficients and κ is the thermal diffusivity coefficient. The diffusion flux represents the diffusion of heat due to temperature gradients. Therefore, the above equation can be written in the form of Fourier's law of heat conduction, i.e., $F_D=-k\nabla T$, where k is the thermal conductivity coefficient and T is the temperature.

The net flux into the finite control volume also consists of the heat sources (absorption or emission of radiation), which include the time rate of heat transfer per unit mass $\dot{q}_h$ and the rate of work done by the body forces f_e. Then the volumetric sources

$$Q_v=\rho\overrightarrow{f_e}\overrightarrow{V}+\dot{q}_h. \quad (43)$$

The surface source Q_s corresponds to the time rate of work done by the pressure and shear normal stresses on the fluid element,

$$Q_s=-p\overrightarrow{V}+\overline{\overline{\tau}}\cdot\overrightarrow{V}. \quad (44)$$

By summing the above contributions we obtain the energy conservation equation,

$$\begin{aligned}\frac{\partial}{\partial t}\int\limits_{\Omega}\rho E d\Omega+\oint\limits_{\partial\Omega}\rho E\left(\overrightarrow{V}\cdot\overrightarrow{n}\right)dS&=\oint\limits_{\partial\Omega}k\left(\nabla T\cdot\overrightarrow{n}\right)dS\\+\int\limits_{\Omega}\left(\rho f_e\cdot\overrightarrow{V}+\dot{q}_h\right)d\Omega-\oint\limits_{\partial\Omega}p\left(\overrightarrow{V}\cdot\overrightarrow{n}\right)dS&+\oint\limits_{\partial\Omega}\left(\overline{\overline{\tau}}\cdot\overrightarrow{V}\right)\cdot\overrightarrow{n}dS.\end{aligned} \quad (45)$$

By applying the relation for the enthalpy $H = h + {}^{V^2}/_2 = E + {}^{p}/_{\rho}$ and gathering the convective $\rho E\overrightarrow{V}$ and pressure term $\left(p\overrightarrow{V}\right)$, we can finally get the energy equation,

$$\begin{aligned}&\frac{\partial}{\partial t}\int\limits_{\Omega}\rho E d\Omega + \oint\limits_{\partial\Omega}\rho H\left(\overrightarrow{V}\cdot\overrightarrow{n}\right)dS = \\ &\oint\limits_{\partial\Omega}k\left(\nabla T\cdot\overrightarrow{n}\right)dS + \int\limits_{\Omega}\left(\rho f_e\cdot\overrightarrow{V}+\dot{q}_h\right)d\Omega + \oint\limits_{\partial\Omega}\left(\overline{\overline{\tau}}\cdot\overrightarrow{V}\right)\cdot\overrightarrow{n}\,dS.\end{aligned} \tag{46}$$

In order to get a better overview of the various terms involved, the three conservation laws can be collected into one system. For this purpose let define two flux vectors: F_c as the vector of convective fluxes and F_v as the vector of viscous fluxes. In addition, the source term Q comprises all volume sources due to body forces and volumetric heating. By casting equations (34, 35, 36) we get

$$\frac{\partial}{\partial t}\int\limits_{\Omega}\overrightarrow{W}d\Omega + \oint\limits_{\partial\Omega}\left(\vec{F}_c - \vec{F}_v\right)dS = \int\limits_{\Omega}\overrightarrow{Q}d\Omega. \tag{47}$$

6.4 The full potential equation (FPE)

The derivation of the FPE will be presented in this section. The approach taken here to derive the FPE is one of several possible ways. More complete information on the derivation can be found in [12].

The equation of mass conservation in a steady state fluid flow can be written in divergence form,

$$\nabla\cdot(\rho\overrightarrow{V}) = 0, \tag{48}$$

where $\overrightarrow{V} = (u, v)^T$ is the velocity and ρ is the local density. Since we are dealing with a potential flow, by definition the flow is irrotational, which implies that there exists a velocity potential such that

$$\overrightarrow{V} = \nabla\phi. \tag{49}$$

The density is related to the velocity (and hence to the potential ϕ) through the energy equation. Let us start from the u-momentum and v-momentum equation

$$\frac{d}{dx}(\rho u^2 + p) + \frac{d}{dy}(\rho uv) = 0, \tag{50}$$

or, applying the derivatives in x and y directions, we get

$$\rho u\frac{d}{dx}(u) + u\left[\frac{d}{dx}(\rho u) + \frac{d}{dy}(\rho v)\right] + \rho u\frac{d}{dy}(v) + \frac{d}{dx}(p) = 0. \tag{51}$$

The terms within the square brackets equal zero because of the continuity equation. Thus the u-momentum equation becomes

$$\frac{d}{dx}\left[\frac{(u^2+v^2)}{2} + \frac{1}{\rho}\frac{d}{dx}(p)\right] = 0. \tag{52}$$

In deriving the above equation we made use of the irrationality relationship $(\nabla \times \overrightarrow{V} = 0)$, $\frac{d}{dy}(u) = \frac{d}{dx}(v)$.

Furthermore, for isentropic flow of a perfect gas, the relation $\frac{p}{\rho^\gamma} = const$ or $\frac{\rho}{\rho_\infty} = \left(\frac{p}{p_\infty}\right)^{\frac{1}{\gamma}}$ holds. Thus,

$$\frac{1}{\rho}\frac{d}{dx}(p) = \frac{\gamma}{\rho_\infty}\frac{(p_\infty)^{\frac{1}{\gamma}}}{\gamma - 1}\frac{d}{dx}\left(p^{(1-\frac{1}{\gamma})}\right). \tag{53}$$

Therefore, the u-momentum becomes

$$\frac{d}{dx}\left[\left(\frac{u^2+v^2}{2}\right) + \frac{\gamma}{\gamma-1}\frac{(p_\infty)^{\frac{1}{\gamma}}}{\rho_\infty}p^{(1-\frac{1}{\gamma})}\right] = 0, \tag{54}$$

or

$$\frac{(u^2+v^2)}{2} + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = G(y), \tag{55}$$

where $G(y)$ is an arbitrary function of y or a constant in x. Doing similar manipulations with the v-momentum equation, we get

$$\frac{(u^2+v^2)}{2} + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = F(x), \tag{56}$$

where $F(x)$ is an arbitrary function x. Comparing (55) and (56) we conclude that $F(x) = G(y) = Constant$. Thus, the energy equation becomes

$$\frac{(u^2+v^2)}{2} + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = const. \tag{57}$$

At the free-stream the above equation becomes

$$\frac{(u_\infty^2+v_\infty^2)}{2} + \frac{\gamma}{\gamma-1}\frac{p_\infty}{\rho_\infty} = const. \tag{58}$$

Comparing (57) and (58) and also noting that the sound speed is defined as $a^2 = \gamma\frac{p}{\rho}$, we get the relation between the local speed of sound and the flow speed:

$$a^2 = a_\infty^2 + \frac{\gamma-1}{2}\left(V_\infty^2 - u^2 - v^2\right). \tag{59}$$

Comparing (57) and (58) and also noting that $q = \sqrt{(u^2+v^2)}$, and that the sound speed is defined as $\gamma\frac{p_\infty}{\rho_\infty} = a_\infty^2 = \frac{V_\infty}{M_\infty^2} = 1$, we get

$$\frac{(u^2+v^2)}{2} + \frac{1}{M_\infty^2(\gamma-1)}\rho^{(\gamma-1)} = \frac{1}{2} + \frac{1}{M_\infty^2(\gamma-1)}. \tag{60}$$

The density can be referenced to a uniform free stream at infinity, or to a stagnation point condition. We derive the free stream version,

$$\rho(\phi) = \rho_\infty\left[\frac{(\gamma-1)}{2}\left(V_\infty^2 - \phi_x^2 - \phi_y^2\right)\right]^{\frac{1}{(\gamma-1)}}. \tag{61}$$

By applying the above relations, the five unknown fields u, v, p, and ρ are eliminated in favor of the potential function ϕ, which solves the full potential equation:

$$\nabla^2 \left[\rho\left(\phi\right) \nabla\left(\phi\right)\right] = 0. \tag{62}$$

The limitation of the potential flow model – The potential flow model assumes a constancy of the entropy and total enthalpy, $S = const.$, $H = const.$, and $\overrightarrow{\nabla} \times \overrightarrow{V} = 0$, where ϕ is the solution of the mass conservation equation. This model ensures that the momentum and energy conservation laws are satisfied. However, in presence of shock waves, this will no longer be the case since the Rankine-Hugoniot relations lead to an entropy increase through a shock. If the shock intensity is uniform the flow remains irrotational. If the shock intensity is not constant, the flow is not irrotational and hence the potential flow model is not "good" anymore, since the potential model implies constant entropy and has therefore no mechanisms to generate entropy variations over the shock [12]. In this work it will be assumed that any shock waves in the flow are weak enough so that the entropy and vorticity generated by the shock can be ignored without introducing serious errors.

6.5 Transonic flow - model problem formulation

Transonic flow is a flow in which large subsonic and supersonic regions coexist. Usually, the supersonic region of the flow is bounded by sonic lines with smooth gradual acceleration of the flow from subsonic to supersonic, and by shock waves through which the flow slows to subsonic speeds. This type of flow occurs in a variety of applications such as flow over aircraft wings, helicopter rotor blades, flow inside compressors and turbines, inlets. Starting from the mass conservation written explicitly, we get

$$\frac{\partial \rho}{\partial x} u + \rho \frac{\partial u}{\partial x} + \frac{\partial \rho}{\partial y} v + \rho \frac{\partial v}{\partial y} = 0. \tag{63}$$

The terms $\frac{\partial \rho}{\partial x}$ and $\frac{\partial \rho}{\partial y}$ can be evaluated by using the momentum equations in x and y directions:

$$\begin{aligned} u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} = -\frac{a^2}{\rho}\frac{\partial \rho}{\partial x} \\ u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} = -\frac{a^2}{\rho}\frac{\partial \rho}{\partial y} \end{aligned}, \tag{64}$$

since $a^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s=const.}$, assuming the entropy is constant. Finally, the FPE in the quasi-linear form is obtained,

$$(a^2 - u^2)\phi_{xx} - 2uv\phi_{xy} + (a^2 - v^2)\phi_{yy} = 0 \tag{65}$$

where ϕ is the potential, u and v are the velocity components, and a is the speed of sound. This equation is a nonlinear second order partial differential equation for ϕ, of the form

$$A\phi_{xx} + B\phi_{xy} + C\phi_{yy} = D, \tag{66}$$

where the two characteristics directions are given by

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A}. \tag{67}$$

Taking the coefficients in (65) the characteristics are

$$\frac{dy}{dx} = \frac{-\phi_x \phi_y \pm \sqrt{a^2 \left(q^2 - a^2\right)}}{a^2 - \phi_x^2}. \tag{68}$$

The characteristics of the FPE are the edges of the Mach cone and are symmetric about the velocity vector $\vec{V}$. The discriminant will determine the nature of the equation. When $q < a$ the characteristics are imaginary and the equation is elliptic. As $q > a$ the discriminant is positive, which results in real values for the roots and therefore real characteristics – the equation becomes hyperbolic. Any numerical scheme used to solve the FPE for ϕ has to reflect the domain of dependence in both the subsonic and supersonic flow regimes.

The precise manner in which both x and y derivatives are shifted in supersonic regions was discovered by Jameson (see [27]), which is called the "rotated difference scheme". In this method the x and y derivatives are shifted correctly so that the numerical and physical domains of dependence coincide, regardless of the local flow direction. The first step towards the rotated difference scheme is to re-express the quasi-linear form (see Eq. (65)) in a new Cartesian coordinate system, $s-n$, where the s-axis is aligned with the flow direction and n is normal to it, as illustrated in Figure 8.

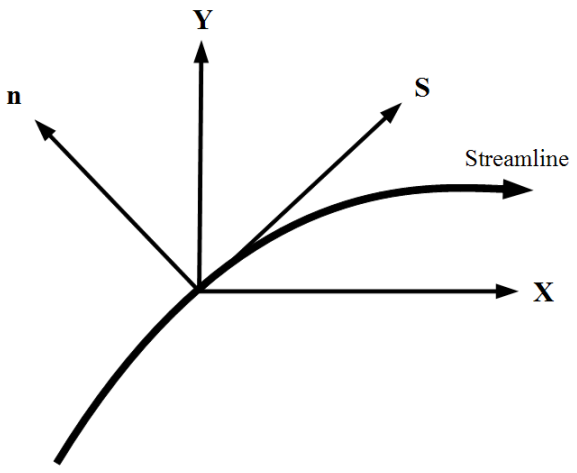


Figure 8: Cartesian coordinate system $s - n$. The s-axis is aligned with the flow direction, and n-axis is normal to it.

Denote the angle between the x-axis and s-axis, equal to the flow angle, θ. Coordinates $s-n$ and $x-y$ are related by the following transformation:

$$\begin{array}{ll} x = s\cos(\theta) - n\sin(\theta) & s = x\cos(\theta) + y\sin(\theta) \\ y = s\sin(\theta) + n\cos(\theta)\,, & n = y\cos(\theta) - x\sin(\theta)\,. \end{array} \tag{69}$$

The derivatives with respect to x and y in terms of s and n are computed as follows:

$$\begin{aligned} \frac{\partial}{\partial x} &= \left(\frac{\partial}{\partial s}\right)\frac{\partial s}{\partial x} + \left(\frac{\partial}{\partial n}\right)\frac{\partial n}{\partial x} = \left(\frac{\partial}{\partial s}\right)\cos(\theta) - \left(\frac{\partial}{\partial n}\right)\sin(\theta) \\ \frac{\partial}{\partial y} &= \left(\frac{\partial}{\partial s}\right)\frac{\partial s}{\partial y} + \left(\frac{\partial}{\partial n}\right)\frac{\partial n}{\partial y} = \left(\frac{\partial}{\partial s}\right)\sin(\theta) + \left(\frac{\partial}{\partial n}\right)\cos(\theta)\,. \end{aligned} \tag{70}$$

The second order derivatives are:

$$\begin{aligned} &\frac{\partial}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\right) = \left[\left(\frac{\partial}{\partial s}\right)\cos(\theta) - \left(\frac{\partial}{\partial n}\right)\sin(\theta)\right]\left[\left(\frac{\partial}{\partial s}\right)\cos(\theta) - \left(\frac{\partial}{\partial n}\right)\sin(\theta)\right] \\ &= \cos^2(\theta)\frac{\partial^2}{\partial s^2} - 2\cos(\theta)\sin(\theta)\frac{\partial^2}{\partial s\partial n} + \sin^2(\theta)\frac{\partial^2}{\partial n^2} \\ &\frac{\partial}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial}{\partial y}\right) = \left[\left(\frac{\partial}{\partial s}\right)\sin(\theta) + \left(\frac{\partial}{\partial n}\right)\cos(\theta)\right]\left[\left(\frac{\partial}{\partial s}\right)\sin(\theta) + \left(\frac{\partial}{\partial n}\right)\cos(\theta)\right] \\ &= \cos^2(\theta)\frac{\partial^2}{\partial n^2} + 2\cos(\theta)\sin(\theta)\frac{\partial^2}{\partial s\partial n} + \cos^2(\theta)\frac{\partial^2}{\partial s^2} \\ &\frac{\partial}{\partial x\partial y} = \left[\left(\frac{\partial}{\partial s}\right)\cos(\theta) - \left(\frac{\partial}{\partial n}\right)\sin(\theta)\right]\left[\left(\frac{\partial}{\partial s}\right)\sin(\theta) + \left(\frac{\partial}{\partial n}\right)\cos(\theta)\right]. \end{aligned} \tag{71}$$

By applying $\cos(\theta) = {}^{u}/_{q}$ and $\sin(\theta) = {}^{v}/_{q}$, Eq. (65) becomes

$$(a^2 - q^2)\phi_{ss} + a^2\phi_{nn} = 0. \tag{72}$$

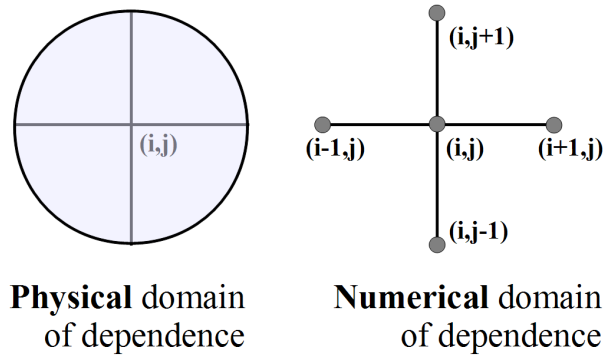
It can be seen that in subsonic regions, when $a > q$, Eq. (72) is elliptic, and hyperbolic in supersonic regions. Now we have to identify the terms in the quasilinear form that contribute to ϕ_{ss} and ϕ_{nn}, which are easily found from (71) as follows:

$$\begin{aligned} &\frac{\partial}{\partial s} = \frac{\partial}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial}{\partial y}\frac{\partial y}{\partial s} = \cos(\theta)\frac{\partial}{\partial x} + \sin(\theta)\frac{\partial}{\partial y}, \\ &\frac{\partial^2}{\partial s^2} = \frac{\partial}{\partial s}\left(\frac{\partial}{\partial s}\right) = \left[\cos(\theta)\frac{\partial}{\partial x} + \sin(\theta)\frac{\partial}{\partial y}\right]\left[\cos(\theta)\frac{\partial}{\partial x} + \sin(\theta)\frac{\partial}{\partial y}\right], \\ &= \cos^2(\theta)\frac{\partial^2}{\partial x^2} + 2\sin(\theta)\cos(\theta)\frac{\partial^2}{\partial x\partial y} + \sin^2(\theta)\frac{\partial^2}{\partial y^2}, \\ &\frac{\partial}{\partial n} = \frac{\partial}{\partial x}\frac{\partial x}{\partial n} + \frac{\partial}{\partial y}\frac{\partial y}{\partial n} = -\sin(\theta)\frac{\partial}{\partial x} + \cos(\theta)\frac{\partial}{\partial y}, \\ &\frac{\partial^2}{\partial n^2} = \left[-\sin(\theta)\frac{\partial}{\partial x} + \cos(\theta)\frac{\partial}{\partial y}\right]\left[-\sin(\theta)\frac{\partial}{\partial x} + \cos(\theta)\frac{\partial}{\partial y}\right], \\ &= \sin^2(\theta)\frac{\partial^2}{\partial x^2} - 2\sin(\theta)\cos(\theta)\frac{\partial^2}{\partial x\partial y} + \cos^2(\theta)\frac{\partial^2}{\partial y^2}, \end{aligned} \tag{73}$$

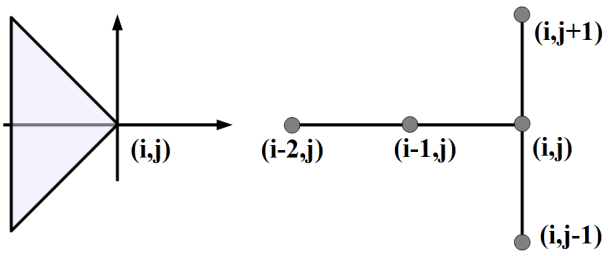
where ϕ_{ss} and ϕ_{nn} are

$$\begin{aligned}\phi_{ss} &= \cos^2(\theta)\phi_{xx} + 2\sin(\theta)\cos(\theta)\phi_{xy} + \sin^2(\theta)\phi_{yy},\\ \phi_{nn} &= \sin^2(\theta)\phi_{xx} - 2\sin(\theta)\cos(\theta)\phi_{xy} + \cos^2(\theta)\phi_{yy}.\end{aligned} \tag{74}$$

The transonic flow problem is a very difficult one from the computational standpoint since Eq. (72) changes type. It is elliptic in the subsonic region, giving rise to signals propagating in all directions. Every point in the domain will be influenced by every other point. As the flow velocity nears the sonic condition, a point in the flow will be more affected by the upstream. In the supersonic region the equation becomes hyperbolic, and the information signals will travel only in the flow direction. Until 1971 the attempts of scientists to solve the transonic flow problem numerically were successful only in the subsonic flow regime. Earl Murman and Julian Cole [20] were the first to recognize the fact that since the equation changes its character from elliptic to hyperbolic, different discreet operators should be applied for each of these two cases. Murman and Cole observed that the dependence of the point (i, j) on the entire neighborhood is physically correct only in the subsonic case where the PDE is elliptic (see Figure 9). Murman and Cole recognized that in the supersonic regions, since the domain of dependence of a certain grid point (i, j) is bounded by the characteristics, the numerical scheme should reflect this upstream dependency. Therefore, they proposed to use the upwind scheme (see Figure 9(b)).



(a)



(b)

Figure 9: a.) In elliptic regions, the point (i, j) depends on the surrounding region. b.) In supersonic regions, the point (i, j) should depend primarily on the information within the characteristic cone.

The small disturbance equation was solved in this way, where the flow was aligned with the x-axis. The solution was applied as follows — a line Gauss–Seidel relaxation along the y-axis grid lines in the order that follows the flow direction. In the present work we insist in using a pointwise relaxation (for the reasons stated previously). If the flow is subsonic (and central differencing is used), a simple pointwise relaxation (Gauss–Seidel or damped Jacobi) is stable. In the supersonic case, however (when an upwind difference approximation is used), such a relaxation is unstable (the amplification factor is greater than unity for certain components). The main achievement of the work presented in this paper is the development of a relaxation procedure that is stable when applied in the pointwise manner.

6.6 Subsonic flow — the discrete approximation

Assume the equation is to be discretized on a uniform Cartesian grid while $\Delta x = \Delta y = h$. Also recall that the aim of this work is not only to construct a stand-alone full potential equation solver but also a building block for a future algorithm for solving the full flow equations based on the factorizable discretization methods (see [50]). Therefore, when constructing a finite difference approximation to ((72)) we apply the following rationale: the discrete form of this equation is expected to follow from the form of the full potential co factor when the system of Euler equations is discretized by means of the factorizable scheme. Therefore, the concepts of "narrow" and "wide" approximations to derivatives (see [50]) are applied here as well.

A convenient way to express graphically a local discrete operator is through its stencil. A standard way to represent a stencil is by a matrix (see, for instance, [55]).

1. When the flow is grid aligned (the s-axis coincides with the x-axis), the streamwise derivative is approximated by the "wide" central second difference

$$\phi^h_{ss} = \frac{1}{4h^2} \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix}. \tag{75}$$

   The cross-flow second derivative is approximated by the standard "narrow" difference

$$\phi^h_{nn} = \frac{1}{h^2} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}. \tag{76}$$

2. In the case of zero Mach number flow, an approximation to the entire equation is a discrete Laplacian of the following form:

$$\Delta^h \phi = \frac{1}{4h^2} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{77}$$

3. It follows from here that the concepts of "wide" and "narrow" second finite differences have to be generalized to the arbitrary direction.

First, we generalize a "narrow" second finite difference to a general direction. Consider the following finite difference stencils:

$$\begin{aligned}
\phi^s_{xx} = \phi^n_{xx} &= \tfrac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \\
\phi^s_{yy} = \phi^n_{yy} &= \tfrac{1}{h^2}\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \\
\phi^s_{xy} &= \tfrac{1}{2h^2}\begin{bmatrix} 0 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \\
\phi^n_{xy} &= \tfrac{1}{2h^2}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}.
\end{aligned} \tag{78}$$

Then the "narrow" second difference in direction n can be given by the following expression

$$\phi^h_{nn} = \sin^2(\theta)\phi_{xx} + 2\sin(\theta)\cos(\theta)\phi^n_{xy} + \cos^2(\theta)\phi_{yy}. \tag{79}$$

Note, that this difference approximates the cross-flow second derivative. Also it is sufficient for the purpose of this work to consider the case where the flow direction forms with, say, the x-axis an angle

$$0^o \le \theta \le 45^o. \tag{80}$$

Therefore, the expression (79) covers all the relevant situations. The "wide" difference in general flow direction is defined simply by subtracting ((79)) from the "wide" approximation to Laplacian ((77)).

6.7 Supersonic flow

In the supersonic region the equation changes type from elliptic to hyperbolic, and, therefore upwind differencing should be used to approximate the second derivative in the flow direction ϕ_{ss} (see ((72))). Again, in the case of grid aligned flow, the second derivative in the flow direction is going to be approximated by *a one sided* "wide" second difference. In the case of a general flow direction, the following approximations to the derivatives are employed:

$$\partial^s_{xx} = \frac{1}{h^2}\begin{bmatrix} \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & \left(-\frac{1}{2} + \sin^2(\theta)\right) & \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & 0 & 0 \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 & 0 \\ \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & \left(-\frac{1}{2} + 2\sin^2(\theta)\right) & \left(\frac{1}{4} - \frac{5}{2}\sin^2(\theta)\right) & sin^2(\theta) & 0 \end{bmatrix}, \tag{81}$$

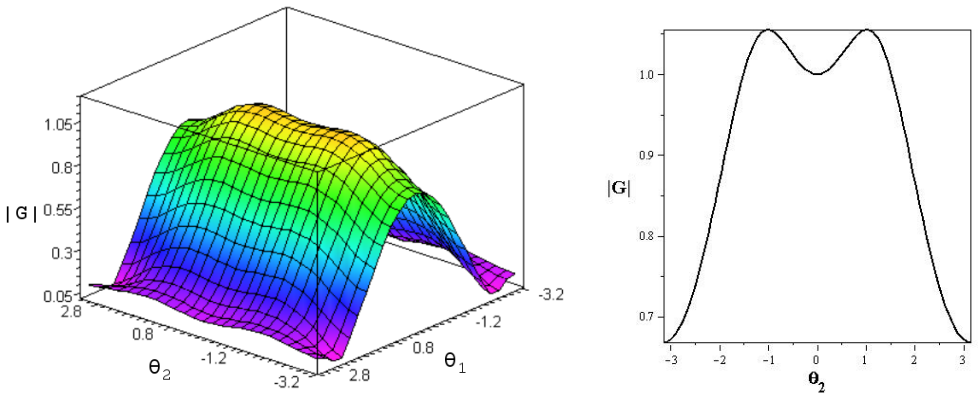
$$\partial^s_{yy} = \frac{1}{h^2}\begin{bmatrix} 0 & \sin^2(\theta) & 0 & 0 & 0 \\ 0 & \left(\frac{1}{4} - \frac{5}{2}\sin^2(\theta)\right) & \frac{1}{2} & \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & 0 \\ 0 & \left(-\frac{1}{2} + 2\sin^2(\theta)\right) & -1 & \left(-\frac{1}{2} + \sin^2(\theta)\right) & 0 \\ 0 & \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & \frac{1}{2} & \left(\frac{1}{4} - \frac{1}{2}\sin^2(\theta)\right) & 0 \end{bmatrix}, \tag{82}$$

$$\partial^s_{xy} = \frac{1}{h^2}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix}. \tag{83}$$

The approximations to ϕ_{nn} remain the same as in the subsonic case.

6.8 Devising a stable relaxation procedure

As stated above, we restrict ourselves in this work to usage of a pointwise relaxation. Therefore, we have to make sure there is a variant of such a relaxation at our disposal that not only is stable for all the cases of interest but also provides a good smoothing. While simple damped Jacobi and symmetric Gauss–Seidel relaxation schemes are suitable for the subsonic case, both of them are unstable in the supersonic case. This can be easily verified by Von–Neumann analysis. As an example, a view of the amplification factor, $|G(\theta_1, \theta_2)|$, as a surface over the region $[-\pi, \pi] \times [-\pi, \pi]$ is given in Figure 10. The amplification factor is a function that describes how the error amplitudes evolve, or, how the amplitude of a mode changes with each relaxation sweep. For convergence of the method, we must have an amplification factor of the damped Jacobi relaxation lower than unity for all the frequencies θ_1 and θ_2, and in this case, this condition is not satisfied. It can be seen that the amplification factor decreases in magnitude as the modes become more oscillatory (θ_1 and θ_2 are increased), but at the slower modes (at the center of the graph) it increases and reaches values exceeding unity.



(a) (b)

Figure 10: a) Amplification factor $|G(\theta_1, \theta_2)|$ for the damped-Jacobi method applied to the discrete upwind operator in two dimensions, shown as a surface over the region $[-\pi, \pi] \times [-\pi, \pi]$. b) Same amplification factor shown as a curve of fixed θ_1.

Several directions were explored to devise a stable pointwise relaxation. One of them was to rely on the well-known decomposition of the supersonic operator into two advection operators, discretizing each one by a certain upwind scheme. A simple pointwise relaxation is stable when applied to these discretizations. However, the overall procedure would suffer from some deficiencies. The main one is that it cannot be generalized to a three-dimensional case. Therefore, this possibility was rejected. Yet another attempt was based on the artificial time method that was introduced by Jameson in [27], while the idea was to use a three-level difference scheme. This approach was also rejected for reasons related to implementation.

6.9 Our approach

A well-known approach that can help in the situation described above is the Kacsmarz relaxation [85, 55]. According to the theory, this relaxation always gives smoothing (provided the operator is not semi-definite, i.e., has no eigenvalues with zero real part), but it is rather inefficient. Instead of solving the algebraic linear system $Ax = b$ directly, it suggests to solve

$$AA^T y = b, \tag{84}$$

where

$$x = A^T y. \tag{85}$$

The implementation of the Kaczmarz relaxation in our case turns out to be quite expensive: since the A matrix has to reflect the upwind (second-order) difference operator, matrix A^T will have to reflect downwind differencing. Their product, therefore, is going to be rather cumbersome. Therefore, our approach was to find a simpler matrix, $\tilde{A}$, to replace A^T and to solve a system $A\tilde{A}y = b$ where $x = \tilde{A}y$. The difference operator resulting in matrix $\tilde{A}$, for the cases of $0^o \leq \theta \leq 45^o$ was chosen to be of the following form:

$$\begin{aligned}
\tilde{L} = &\left(\frac{1}{4}\sin^2(\theta) + \frac{1}{4}\sin(\theta)\cos(\theta)\right)\phi_{i-1,j+1} \\
&+ \left(\frac{1}{4} - \frac{1}{2}\sin(\theta)\cos(\theta)\right)\phi_{i+1,j+1} \\
&+ \frac{1}{2}\sin^2(\theta)\phi_{i,j+1} \\
&+ \left(\frac{1}{2} - \frac{1}{2}\sin^2(\theta)\right)\phi_{i+1,j} \\
&+ \left(\frac{1}{4}\cos^2(\theta) + \frac{1}{4}\sin(\theta)\cos(\theta)\right)\phi_{i+1,j-1} - \phi_{i,j}.
\end{aligned} \tag{86}$$

For illustration, we present this operator in two special cases. When the flow is grid aligned, $\tilde{L}$ is given as follows:

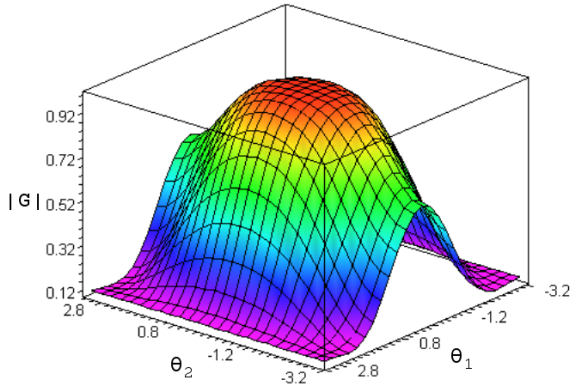
$$\tilde{L} = \frac{1}{h}\begin{bmatrix} 0 & 0 & \frac{1}{4} \\ 0 & -1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} \end{bmatrix}, \tag{87}$$

while for the diagonal direction flow, the discrete operator is

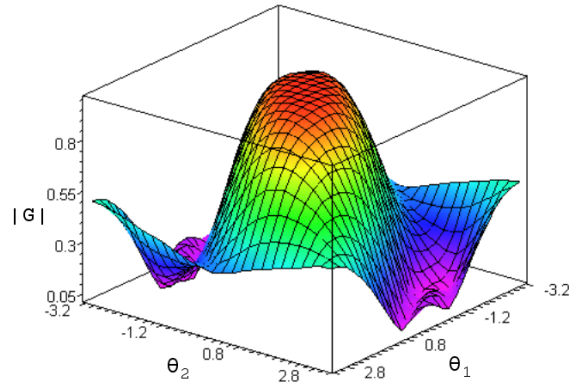
$$\tilde{L} = \frac{1}{h}\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & -1 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \end{bmatrix}. \tag{88}$$

It is important to note that the performance of the overall AMG algorithm depends strongly on the choice of operator $\tilde{L}$. The particular structure presented in (86) seems to lead to the best results. The product operator $L\tilde{L}$ is h-elliptic,

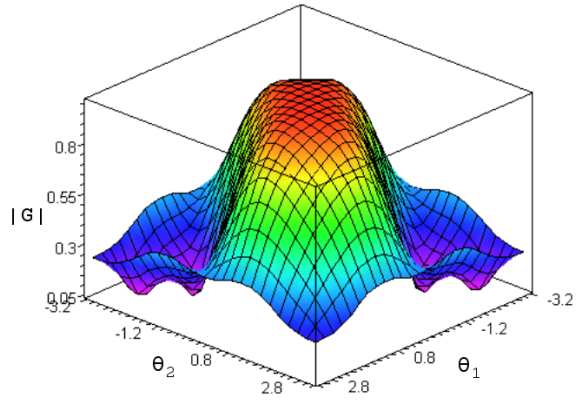
according to the concept of h-ellipticity introduced by Brandt [63]. Figure 11 presents the amplification factor (the function that describes how the error amplitudes evolve - for convergence of the method, we must have an amplification factor lower than unity) for the Gauss–Seidel relaxation method, as a surface over the frequencies $[-\pi, \pi] \times [-\pi, \pi]$. The flow speed for the three cases is $M_\infty = 1.1$ and the flow direction $\theta = 0°, 30°$, and 45^0, respectively. One can see that the whole surface is lower than unity, which indicates that the method is stable. Clearly, the amplification factor decreases as the modes become more oscillatory. The smoothing properties of such a relaxation are much better than those of the classical Kaczmarz (for comparison see Figure 12). To the best of our knowledge, there are no other known results of an application a simple pointwise relaxation and of the AMG method based on it for the transonic flow problem.



(a)



(b)



(c)

Figure 11: Amplification factor , $|G(\theta_1, \theta_2)|$, for the Gauss–Siedel method, applied to the discrete upwind operator in two dimensions, shown as a surface over the region $[-\pi, \pi] \times [-\pi, \pi]$. a) $\theta = 0°$, b) $\theta = 30°$, c) $\theta = 45°$.

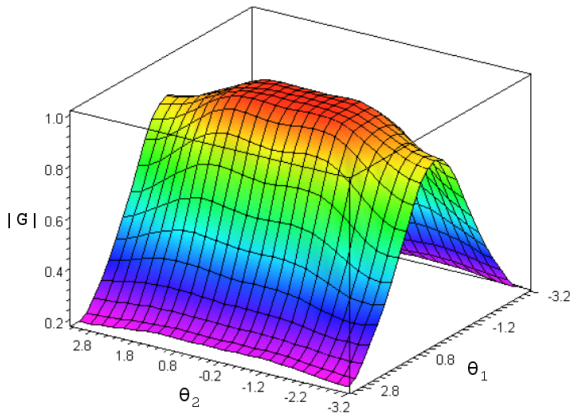


Figure 12: Amplification factor , $|G(\theta_1, \theta_2)|$, applied to the discrete upwind operator in two dimensions, shown as a surface over the region $[-\pi, \pi] \times [-\pi, \pi]$ for the Kaczmarz method. The flow conditions are $M = 1.1$ and $\theta = 0°$.

Now, when we have stable and efficient relaxation schemes for cases of subsonic, sonic, and supersonic flow, it is necessary to devise a single unified methodology that covers all the cases. When the flow is subsonic, a central difference is used for the second derivative in the flow direction, while pointwise relaxation can be applied directly in conjunction with the matrix A (or operator L). If the flow is supersonic, we apply relaxation directly with the product of downwind and upwind operators $L\tilde{L}$ (or matrix $A\tilde{A}$). Since the operator $\tilde{L}$ cannot suddenly appear in the supersonic flow regime, it must exist also in the subsonic flow regime. Therefore, these two schemes have to be combined. One way of doing this is still to keep $\tilde{A}$ in the subsonic case while modifying it so that it gradually becomes a unity matrix as the flow speed decreases.

Once we have a stable relaxation procedure based upon the combined operator, we can proceed to devising the overall solver. However, when operator L is nonlinear it becomes complicated to construct and apply such a combined operator. For the purpose of overcoming this difficulty, we devised and applied the distributive relaxation technique such that it is identical to the simple pointwise relaxation on the operator $L\tilde{L}$ in the linear case.

6.10 Residual distributive relaxation

Denote the residual of the discrete equation at point

$$r = f - L\phi^c, \tag{89}$$

where ϕ^c stands for a current approximation to ϕ. Then (in the case of a linear operator L), the equation for correction can be written as follows

$$L(\delta\phi) = r. \tag{90}$$

However, as discussed previously, a pointwise relaxation procedure applied directly to the above equation is unstable in the hyperbolic case. Therefore, it was

suggested to apply relaxation directly to the "combined" operator

$$L\tilde{L}(\delta y) = r, \tag{91}$$

and, after computing the correction δy, to evaluate the correction $\delta\phi = \tilde{L}(\delta y)$. The entire AMG solution algorithm can address the combined operator $L\tilde{L}$ (or matrix $A\tilde{A}$) and the corresponding unknown y (or correction δy). However, when operating on the finest level, we need to deal with the problem in terms of the unknown ϕ. One way of doing this can be to translate the correction δy into the correction $\delta\phi$ at the end of a relaxation sweep or a multigrid cycle. An alternative way is to perform this translation immediately after computing the correction at each and every point

$$(\delta\phi)_{i,j} = \tilde{L}\,(\delta y)_{i,j}\,. \tag{92}$$

Note, that due to the structure of operator $\tilde{L}$, at grid point i, j we have

$$(\delta\phi)_{i,j} = (\delta y)_{i,j}\,. \tag{93}$$

Therefore, this procedure can be described as evaluating correction $(\delta\phi)_{i,j}$ at a point based on the original operator L. However, in addition to introducing it at point i, j and in order to avoid the instability of such a "direct" relaxation procedure, we also distribute its fractions at certain downstream points according to the operator $\tilde{L}$.

The residual distribution method consists of distributing fractions of the flux balance in a cell to the adjacent cells in the upwind direction, with weights summing up to one for consistency. The idea is to build a procedure that accurately mimics the structure of the operator $\tilde{L}$ in all the flow conditions that we have tested. The distribution of the residual in every point in the field is done in a systematic procedure, exactly according to the structure of the operator $\tilde{L}$.

Consider a subsonic grid aligned flow, in a cell center far from the boundaries. After computing the residual, it is distributed to three adjacent cells according to the operator $\tilde{L}$,

$$\begin{aligned}
\phi_{i-1,j+1} &= \phi_{i-1,j+1} + M_{i,j}^2 \frac{1}{4} r_{i,j}, \\
\phi_{i-1,j} &= \phi_{i-1,j} + M_{i,j}^2 \frac{1}{2} r_{i,j}, \\
\phi_{i-1,j-1} &= \phi_{i-1,j-1} + M_{i,j}^2 \frac{1}{4} r_{i,j},
\end{aligned} \tag{94}$$

where $r_{i,j}$ is the residual computed in the cell (i, j) and $M_{i,j}$ is the local Mach number through the face (i, j). As one can see, the the additional weight to each cell is modified so it gradually becomes zero as the flow speed decreases.

7 Extending the AMG method to transonic flow computations

7.1 The coarsening process by using a dynamic threshold

In the context of an algebraic multigrid we are going to deal at each level with a linear system of equations

$$A^m u^m = f^m, \tag{95}$$

where m is the level index. As mentioned in the previous section, the coarsening process is derived based on the strong and weak connections between unknowns, which essentially measure the relative size of the off-diagonal entries. Connections between neighboring variables are considered strong if the size of the corresponding matrix entry exceeds a certain threshold, relative to the maximum entry of the row. This threshold value is very important for constructing a good coarse-grid. According to [69], a point i is said to be strongly connected to j, if

$$-a_{ij}^m \geq \varepsilon \max_{k \neq i}(-a_{ik}^m). \tag{96}$$

The threshold value ε is kept fixed for most applications, with a typical value of 0.25. It is well-known that devising a robust coarsening process, which results in an accurate interpolation, is one of the keys for achieving a good AMG convergence. There are special cases, for example, anisotropic elliptic problems (anisotropy can be a result of the computational grid or the equation itself), where the equation changes type from elliptic to hyperbolic (transonic flow problem) and a fixed threshold parameter can result in an inadequate coarsening process.

The problem under consideration (supersonic case, for instance) usually leads to a matrix with significant negative off-diagonal entries. Therefore, it is important to redefine the definition of strong and weak connections, so that it becomes more adequate for our case. We would like to allow a connection with a negative coefficient whose absolute value is sufficiently large, to be classified as a strong one. Therefore, as suggested in [76], we modify definition (96): a point i is considered to be strongly connected to point j, if

$$\left|a_{ij}^m\right| \geq \varepsilon \max_{k \neq i} \left|a_{ik}^m\right|. \tag{97}$$

Our further generalization of this idea is to select strong connections using a different threshold parameter for each row of the matrix A^m, while computing its value by analyzing this row's entries. Each row represents a difference operator at a specific grid point and, possibly, boundary conditions. Intuitively speaking, the same fixed threshold value (as used in the classical AMG) cannot reflect the relative size of the row's entries for various cases (isotropic, strongly anisotropic, or hyperbolic). Therefore, in order to maintain a good AMG performance across the entire variety of flow speeds, we propose to compute the threshold adaptively, during the coarsening process, for each row of the matrix.

Assuming the operator A^m is known, we start from the following equation:

$$a_{ii}^m u_i^m + \sum_{j \in N_i^m} a_{ij}^m u_j^m = f_i^m \qquad \forall i \in \Omega^m. \tag{98}$$

Let us define the threshold parameter ε for each row i as:

$$\varepsilon = \frac{\sum_{j \neq i} |a_{ij}| \frac{|a_{ij}|}{\max_{k \neq i} |a_{ik}|}}{\sum_{j \neq i} |a_{ij}|}. \tag{99}$$

As one can see, the threshold parameter is simply the weighted average value in a given row. Several observations relating to the constant coefficient case are in order here. If a periodic boundary condition is applied, the matrix A^m is composed of identical rows and the threshold parameter can be calculated once

for each coarse-level. But, while applying other boundary condition (Dirichlet, Neumann), as for a general problem, it is necessary to calculate the threshold for each row separately. This proposed threshold calculation appears to be more adequate than the fixed threshold used in the standard AMG method. This will be illustrated by the numerical experiments in Section 8. Practical experience has shown that while solving the FPE in the conservation form with finite volume schemes, the advantages of the dynamic threshold is even more dramatic. This area is covered in Section 13.

7.2 The coarse-grid and restriction operators

In the classical approach, suggested by Ruge and Stueben, the restriction operator is defined as the transpose of the interpolation, I_{m+1}^m, I_m^{m+1}. Then the coarse-grid operator is defined by the Galerkin-type algorithm, $A^{m+1} = I_m^{m+1} A^m I_{m+1}^m$. We shall refer to this approach as Algorithm 1. Although this is the simplest way to construct the restriction and coarse-grid operators, it leads to poor convergence, when the matrix A^m is not an M-Matrix.

Alternatively, a second algorithm discussed in [76] is to use direct approximations based on the fine-grid operator A^m to construct A^{m+1} and I_m^{m+1}. Assuming the operator A^m is known, we start from the following equation:

$$a_{ii}^m u_i^m + \sum_{j \in C_i^m} a_{ij}^m u_j^m + \sum_{j \in D_i^m} a_{ij}^m u_j^m \approx f_i^m \qquad i \in C^m. \tag{100}$$

In order to derive the coarse-grid operator A^{m+1}, the terms associated with u_j^m, $j \in D^m$, in the ith equation, $i \in C^m$, should be approximated. The simplest way is to distribute these elements according to their values to their strong connections. The resulting coarse-level operator could not provide accurate correction to an approximate solution in the fine-level, and it leads to poor convergence rates. Alternatively, the terms u_j^m, $j \in D^m$, in the ith equation can be replaced by the jth equation. While this operation cannot eliminate all u_j^m, $j \in D^m$, it can reduce the magnitude of the coefficient. The u_j^m, $j \in D^m$ are eliminated by the interpolation formula if $\left|a_{ij}^m / a_{ii}^m\right| \leq \varepsilon_c$ and the u_j^m, $j \in D^m$ are replaced by means of the jth equation when $\left|a_{ij}^m / a_{ii}^m\right| > \varepsilon_c$, where $\varepsilon_c = 0.001$, i.e.,

$$u_j^m = \begin{cases} \sum\limits_{k \in C_j^m} w_{jk} u_k^m, & if \left|a_{ij}^m / a_{ii}^m\right| \leq \varepsilon_c \\ \frac{f_j^m}{a_{jj}^m} - \frac{1}{a_{jj}^m} \sum\limits_{k \in N_j^m} a_{jk}^m u_k^m, & if \left|a_{ij}^m / a_{ii}^m\right| > \varepsilon_c \end{cases}. \tag{101}$$

The new equations are obtained

$$a_{ii}^{(2)} u_i^m + \sum_{j \in C^m} a_{ij}^{(2)} u_j^m + \sum_{j \in D^m} a_{ij}^{(2)} u_j^m \approx f_i^m - \sum_{j \in D_1^m} \frac{a_{ij}^m}{a_{jj}^m} f_j^m \qquad i \in C^m, \tag{102}$$

where $D_1^m = \left\{ j : \left|a_{ij}^m / a_{ii}^m\right| > \varepsilon_c, \;\; j \in D^m \right\}$. This process is repeated until u_j^m,$j \in D^m$is eliminated from the ith equation. Therefore,

$$a_{ii}^{(L)} u_i^m + \sum_{j \in C^m} a_{ij}^{(L)} u_j^m \approx f_i^m - \sum_{l=1}^{L-1} \sum_{j \in F_l^m} \frac{a_{ij}^{(l)}}{a_{jj}^m} f_j^m \qquad i \in C^m, \tag{103}$$

where $a_{ij}^{(1)} = a_{ij}^m$, $D_l^m = \left\{ j : \left| a_{ij}^{(l)} / a_{ii}^{(l)} \right| > \varepsilon_c, \;\; j \in D^m \right\}$. The coarse-grid operator is defined as $A^{m+1} = (a_{ij}^{(L)})$. The restriction operator I_m^{m+1} is resulted directly from (103),

$$f_i^{m+1} = f_i^m - \sum_{l=1}^{L-1} \sum_{j \in C^m} \frac{a_{ij}^{(l)}}{a_{jj}^m} f_j^m, \qquad i \in \Omega^{m+1}. \tag{104}$$

Note that using this method we get $I_m^{m+1} \neq \left(I_{m+1}^m\right)^T$ (see [76] for more details). This approach works well for all the problems considered in this paper.

7.3 The interpolation operator

Two attempts have been made to construct an interpolation operator that can deal with a matrix that includes positive and negative off-diagonal entries. Stueben in [69] described the following choice for the interpolation operator for the variable $i \in \Omega^m$:

$$a_{ii} e_i + \alpha_i \sum_{k \in C_i} a_{ik}^- e_k + \beta_i \sum_{k \in C_i} a_{ik}^+ e_k = 0. \tag{105}$$

Since the matrix under consideration may include both positive and negative off-diagonal entries, the notation they used is:

$$a_{ij}^- = \begin{cases} a_{ij} & if \;\; (a_{ij} < 0) \\ 0 & if \;\; (a_{ij} \geq 0) \end{cases} \qquad \text{and} \qquad a_{ij}^+ = \begin{cases} 0 & if \;\; (a_{ij} \leq 0) \\ a_{ij} & if \;\; (a_{ij} > 0) \end{cases}. \tag{106}$$

Correspondingly,

$$N_i^- = \left\{ j \in N_i : a_{ij}^h < 0 \right\} \qquad \text{and} \qquad N_i^+ = \left\{ j \in N_i : a_{ij}^h > 0 \right\}. \tag{107}$$

The parameters α_i and β_i are defined as follows:

$$\alpha_i = \frac{\sum_{j \in N_i} a_{ij}^-}{\sum_{k \in C_i} a_{ik}^-} \qquad \text{and} \qquad \beta_i = \frac{\sum_{j \in N_i} a_{ij}^+}{\sum_{k \in C_i} a_{ik}^+}. \tag{108}$$

This leads to the following interpolation weights:

$$w_{ik} = \begin{cases} -\alpha_i & \frac{a_{ik}}{a_{ii}} \qquad \left(k \in C_i^-\right) \\ -\beta_i & \frac{a_{ik}}{a_{ii}} \qquad \left(k \in C_i^+\right) \end{cases} \tag{109}$$

where C_i^- and C_i^+ are negative and positive off-diagonal entries, respectively. This approach for constructing the interpolation operator was implemented for the supersonic flow regimes, and resulted in a slow convergence.

The second attempt to improve the interpolation operator, and hence the overall performance in the transonic flow regime, was the implementation of the interpolation formula described in [76]. In this case, we also did not find any advantage in terms of operator complexity and convergence factor for the problems that we have tested.

Our formulation for the interpolation operator is based on Stueben's approach [69], which was demonstrated to be efficient for the M-matrices. The construction of the interpolation operator is identical to the standard interpolation (described in Section 4) except for the definition of weak/strong connections. Namely, the modified criteria are based on comparing the absolute values of the matrix entries (97) and not their values (96). This interpolation formula is more accurate (especially in the supersonic flow regime) than used in the standard AMG method because unlike the standard approach, it allows sufficiently large negative off-diagonal entries also to be considered as strongly connected points. This is verified by numerical experiments reported in Section 8. It is important to mention that the best results in the sonic flow regime were obtained with the classical interpolation operator $I_{m+1}^m = (I_m^{m+1})^T$. Although we present the AMG setup that results in the best performance, our aim is to construct a uniform interpolation operator that works well for all the cases. That is to say, our modified interpolation works well for the entire range of the flow speed, from low Mach number flow up to transonic and supersonic regimes.

7.4 Smoothing

In this work we insist on using a pointwise relaxation method as a smoother for the AMG algorithm. It is motivated by the following observations:

- Pointwise relaxation methods are relatively simple.
- Pointwise relaxation is desirable in the AMG context since (unlike line relaxation) it releases us from reliance on the grid geometry. Using line relaxation within the geometric multigrid context while the flow is grid aligned is quite acceptable, but it is not clear how to extend it to the general flow direction case.
- While using Jacobi or symmetric Gauss–Siedel methods, the relaxation process is independent of the flow direction, and, therefore, of the problem's geometry.
- Can be parallelized effectively.

7.5 The FMG method in the context of AMG for solving nonlinear problems

When a nonlinear problem is involved, it can happen that many AMG cycles are wasted in order to reach a suitable approximation when the initial condition is far from the exact solution. This situation can be avoided by applying the FMG method that was presented in Section 3. However, the original geometric FMG is based on geometric multigrid cycles in order to improve the initial guess on the fine-level. That is, the coarsening process itself is fixed and this puts particular requirements on the smoothing properties of the smoother used in order to ensure efficient interplay between smoothing and coarse-grid correction. Pointwise relaxation is very efficient for isotropic problems. However, for anisotropic problems pointwise relaxation is effective only in the direction of strong connections. Consequently, other smoothers are required in order to get convergence, if it is possible

at all. In order to have an efficient interplay between smoothing and coarse-grid correction and a robust FMG method that produces a good initial guess for the fine-level, we suggest to replacing the MG V-cycles with an AMG V-cycles.

Figure 13 shows the schedule of the grids in the order in which they are visited. The coarse "base" grid of the FMG sequence are based on the classical coarsening strategies – doubling the mesh size in each direction, that is, by $h \rightarrow 2h$ coarsening. Practically, we solve the problem first on a coarse-grid and then interpolate the solution to the finer level. The interpolated solution serves as an initial approximation for the AMG cycle on the finer level. Formally, such an FMG algorithm, based on four levels, can be described as follows:

1. Initialize $I_h^{2h} f^h \rightarrow f^{2h}$, $I_{2h}^{4h} f^{2h} \rightarrow f^{4h}$, ...
2. Solve the problem on coarsest grid $A^{8h} v^{8h} = f^{8h}$.
3. $I_{8h}^{4h} v^{8h} \rightarrow v^{4h}$.
4. $V_{AMG}^{4h} \left(v^{4h}, f^{4h}\right) \rightarrow v^{4h}$, ν_0 times.
5. $I_{4h}^{2h} v^{4h} \rightarrow v^{2h}$.
6. $V_{AMG}^{2h} \left(v^{2h}, f^{2h}\right) \rightarrow v^{2h}$, ν_0 times.
7. $I_{2h}^{h} v^{2h} \rightarrow v^{h}$.
8. $V_{AMG}^{h} \left(v^{h}, f^{h}\right) \rightarrow v^{h}$, ν_0 times.

Instead of transferring the f^h to the coarse-level by restriction, the original right hand side f is used for the coarsest level. The cycling parameter ν_0 sets the number of AMG V-cycles done at each level. Practical experience has shown that ν_0 depends on the problem. Usually for elliptic problems $\nu_0 = 1$ is sufficient to produce a good initial guess for the fine-level.

It is important to mention that this simple FMG algorithm seems still far from being optimal. However, it already appears to be very instrumental in solving nonlinear problems (as presented in Section 13). We believe there is still much room for optimization, albeit, the results indicate that the algorithm is very efficient even as is.

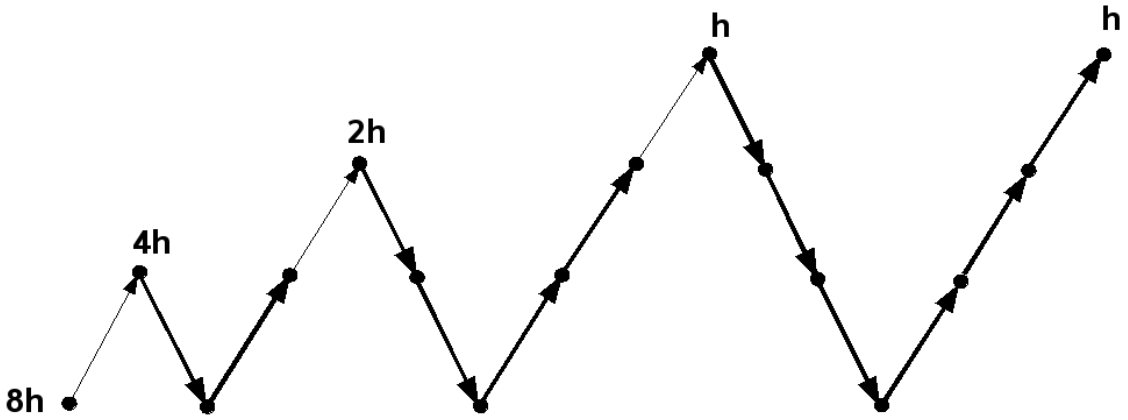


Figure 13: Schedule of grids for the improved FMG scheme on four levels. The bold arrows indicate the AMG V-cycle. The thin arrows present an interpolation of the approximation to the next finer level.

8 Numerical Experiments - Finite Differences

8.1 AMG in subsonic/sonic flow

To evaluate the AMG performance, a series of numerical experiments were performed with isotropic and anisotropic problems with periodic boundary conditions. The symmetric Gauss–Siedel method is employed as a smoother, and the following notation is used for the results presented in the following tables:

cf – convergence factor, defined by

$$cf = \frac{\left\|Au^{(i)} - f\right\|_2}{\left\|Au^{(i-1)} - f\right\|_2}, \tag{110}$$

where $u^{(i)}$denote the ith algebraic multigrid iteration.

C_g – grid complexity – total number of elements, on all the levels, divided by the number of elements on the finest level (as introduced in Section 4).

To give insight into the coarsening process and the structure of the coarser levels, we illustrate them graphically. Figure 14 presents two consecutive levels, fine and coarse, for various Mach numbers and flow directions. The average convergence factor for problems of varying sizes is listed in Table 1. As one can see, it is independent of the problem size. For an isotropic case, say $M = 0$, the proposed coarsening algorithm selects, as expected, an isotropic coarse-grid (the same as would be obtained by the geometric multigrid coarsening). The proposed algorithm appears robust for anisotropic cases as well. For grid aligned anisotropy ($\theta = 0°$), as sketched in Figure 14(b), the coarsening algorithm chooses a grid identical to one that would be obtained by semicoarsening in the geometric multigrid context. For grid diagonal aligned anisotropy case ($\theta = 45°, M = 0.9$), the coarse-grid chosen is shown in Figure 14(c). Recalling that the discrete operator in this case consists of the product $L{\cdot}\tilde{L}$ and due to the setting $\varepsilon = 0.25$, however, AMG treats connections in the x-direction as strong. We conclude that the

obtained coarsening pattern is an excellent choice for this problem and it is well reflected in the convergence rates presented in Table 1. Figure 14(d) represents the result of the coarsening procedure for a general anisotropic case ($\theta = 30^o$).

The average convergence factor and grid complexity for problems of varying sizes are listed in Table 1. As one can see, the convergence rate is bounded independent of the problem size. Solving the problem on reduced mesh sizes maintains the structure of the fine-scale problem and so the grid complexity remains nicely bounded, independent of problem size, with values of approximately $\frac{4}{3}$ for the Laplace problems and 2 for the anisotropic problems.

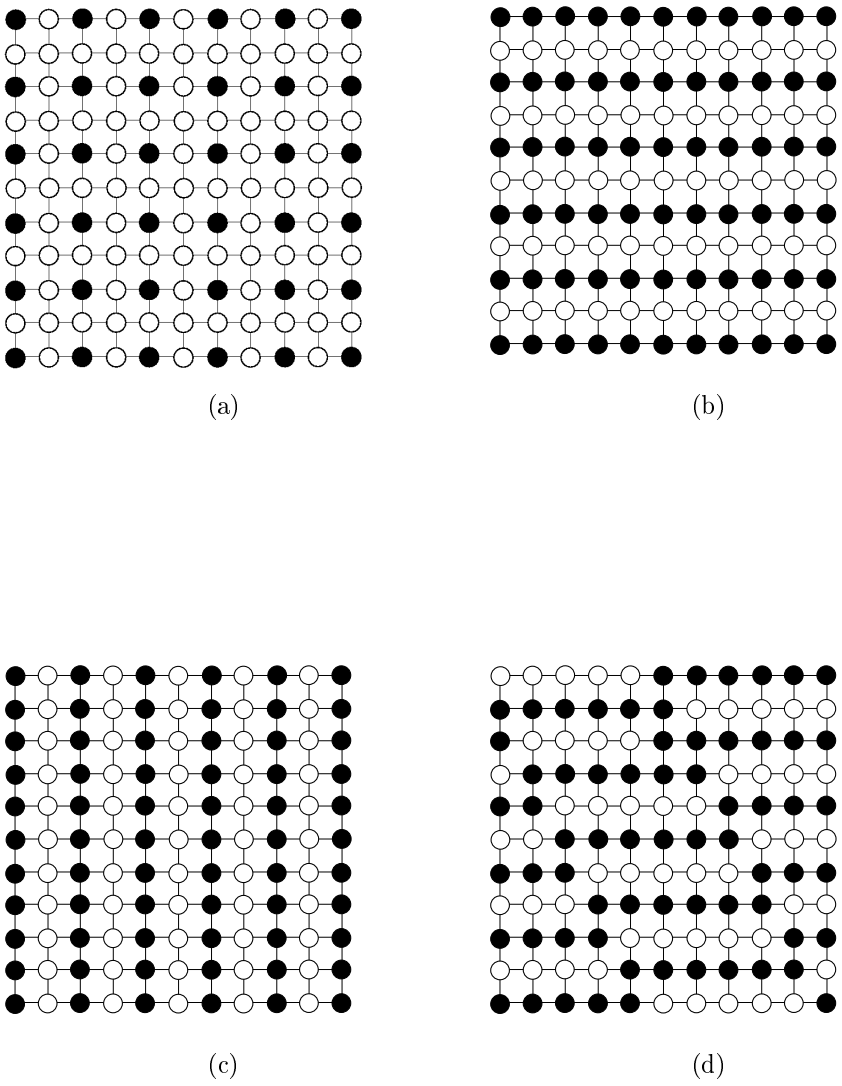


Figure 14: Various coarsening for a nine-point discretization on a 11×11 grid. White points are F-points, black points are C-points. a) $M_\infty = 0$, $\theta = 0^0$. b) $M_\infty = 0.95$, $\theta = 0^0$. c) $M_\infty = 0.9$, $\theta = 45^0$. d) $M_\infty = 0.95$, $\theta = 30^0$.

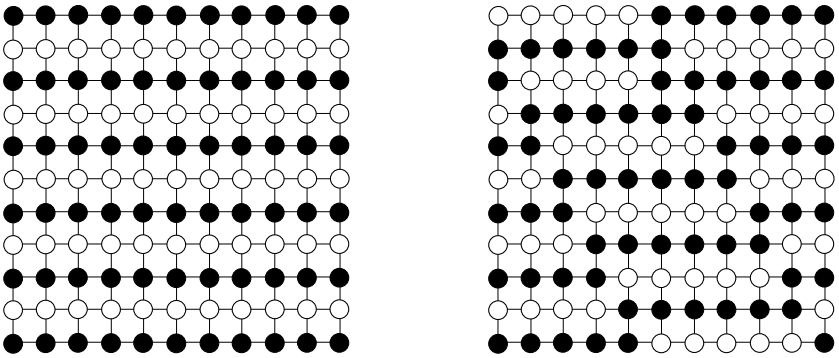
Table 1: Convergence rate C_f and grid complexity C_Ω of resulting $V(2,1)$ AMG cycles.

	Grid	$M_\infty = 0.25$		$M_\infty = 0.5$		$M_\infty = 0.75$		$M_\infty = 0.95$		$M_\infty = 1$	
θ[rad]		C_f	C_Ω	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω
0	25×25	0.04	1.38	0.06	1.43	0.05	1.88	0.06	2.07	0.07	1.96
	50×50	0.05	1.37	0.05	1.38	0.05	1.84	0.06	1.98	0.07	1.96
	100×100	0.05	1.35	0.06	1.35	0.06	1.82	0.07	1.95	0.07	1.96
$\frac{\pi}{6}$	25×25	0.05	1.37	0.05	1.38	0.06	1.44	0.14	1.58	0.08	1.93
	50×50	0.05	1.36	0.05	1.36	0.06	1.42	0.07	1.59	0.08	1.93
	100×100	0.06	1.35	0.06	1.35	0.06	1.38	0.06	1.37	0.08	1.93
$\frac{\pi}{4}$	25×25	0.05	1.38	0.05	1.41	0.08	1.43	0.05	2.02	0.04	1.78
	50×50	0.05	1.36	0.05	1.40	0.05	1.40	0.04	1.92	0.04	1.78
	100×100	0.06	1.35	0.06	1.40	0.07	1.39	0.05	1.9	0.04	1.78

8.2 Application of AMG in supersonic flow

The AMG performance for supersonic flow with various Mach numbers and flow directions, while using SGS as a smoother, is presented in Table 2 and Table 3. Coarsening diagrams for these cases are presented in Figure 15. In all cases the same initial guess was used. Periodic boundary conditions were imposed. Once the setup phase was completed, $V(2,1)$ AMG cycles were applied in the solving phase. The convergence rate is slightly slower here compared to the subsonic flow, but still lower than 0.1 in most cases. As expected, the memory requirements for these strongly anisotropic problems is typically higher than that for isotropic problems (see Table 1). The reason being that AMG essentially performs one-dimensional coarsening in the direction of strong connectivity.

As a comparison we present the AMG performance for the same flow conditions and grid sizes while applying the damped Jacobi relaxation method ($\omega = 0.7$) as a smoother (see Table 3). It is clearly seen that the convergence is significantly slower than that for the SGS smoother and its rate exhibits a slight h-dependence. For higher grid size (100×100) the situation is even worse; no convergence was achieved. The sign (–) indicates the variants of the algorithm that were not convergent).



(a) (b)

Figure 15: Various coarsening for a nine-point discretization on an 11×11 grid. White points are F-points, black points are C-points. a) $M_\infty = 1.05$, $\theta = 0^0$, b) $M_\infty = 1.15$, $\theta = 30^0$.

Table 2: Asymptotic convergence factors of resulting V(2,1) cycles, while applying SGS relaxation method.

			$M_\infty = 1.05$		$M_\infty = 1.1$		$M_\infty = 1.15$		$M_\infty = 1.2$	
	θ[rad]	**Grid**	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω
Fixed thr.	0	25×25	0.16	1.98	0.11	1.95	0.11	2.12	0.18	2.18
		50×50	0.12	1.90	0.17	1.91	0.26	1.94	0.18	2.19
		100×100	0.12	1.90	0.2	1.90	0.27	1.90	0.2	2.20
Dynamic thr.		25×25	0.05	2.23	0.08	2.13	0.07	2.16	0.06	2.20
		50×50	0.08	2.21	0.10	2.07	0.07	2.12	0.06	2.20
		100×100	0.09	2.18	0.10	2.12	0.09	2.08	0.13	2.06
Fixed thr.	$\frac{\pi}{6}$	25×25	0.07	2.02	0.10	1.54	0.11	1.56	0.11	1.57
		50×50	0.15	2.01	0.11	1.54	0.11	1.55	0.1	1.55
		100×100	–	–	0.11	1.54	0.11	1.55	0.18	1.54
Dynamic thr.		25×25	0.06	2.19	0.04	2.20	0.07	2.10	0.04	2.98
		50×50	0.07	2.13	0.07	2.14	0.07	2.10	0.04	3.0
		100×100	0.07	2.12	0.09	2.12	0.1	2.12	0.04	3.0
Fixed thr.	$\frac{\pi}{4}$	25×25	0.11	2.01	0.11	1.55	0.11	1.56	0.13	1.61
		50×50	0.16	2.04	0.11	1.52	0.14	1.52	0.13	1.60
		100×100	0.2	2.02	0.4	1.54	0.14	1.54	0.26	1.57
Dynamic thr.		25×25	0.07	2.11	0.1	2.14	0.06	2.18	0.07	2.14
		50×50	0.08	2.06	0.1	2.08	0.06	2.11	0.07	2.12
		100×100	0.08	2.06	0.06	2.10	0.07	2.11	0.08	2.10

Table 3: Asymptotic convergence factor C_f and grid complexity C_Ω of resulting V(2,1) cycles, while applying damped Jacobi relaxation method ($\omega = 0.7$). In the coarsening process a dynamic threshold was applied.

			$M_\infty = 1.05$		$M_\infty = 1.1$		$M_\infty = 1.15$		$M_\infty = 1.2$	
	θ[rad]	**Grid**	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω	C_f	C_Ω
Dynamic thr.	0	25×25	0.35	2.23	0.30	2.13	0.45	2.17	0.27	2.22
		50×50	0.4	2.21	0.9	2.07	0.8	2.11	0.50	2.20
		100×100	0.4	2.17	–	–	–	–	0.57	2.17
Dynamic thr.	$\frac{\pi}{6}$	25×25	0.47	2.19	0.27	2.21	0.41	2.16	0.33	2.98
		50×50	0.47	2.13	0.28	2.20	0.45	2.12	0.37	3.05
		100×100	0.48	2.12	0.35	2.14	0.74	2.08	0.44	3.05
Dynamic thr.	$\frac{\pi}{4}$	25×25	0.32	2.11	0.30	2.14	0.33	2.18	0.37	2.14
		50×50	0.73	2.06	0.51	2.08	0.67	2.11	–	–
		100×100	–	–	–	–	–	–	–	–

Our observations from these numerical experiments are then as follows:

1. Since a periodic boundary condition is used, the second pass process has to be applied for all the coarse-levels in order to satisfy the interpolation requirements. Both Algorithms 1 (Galerkin algorithm) and 2 (direct approximation) converge. Although Algorithm 1 is simpler to implement, there is a significant advantage to the second method over the first one in the supersonic flow regime, where the discrete operator does not lead to an M-matrix.

2. In the coarsening process while using Algorithm 1, the terms associated with u_j^m, $j \in F^m$, in the ith equation (see (103)), are not totally eliminated. Therefore, the remaining small values can be distributed among the coarse-level points in this row $i \in C_i^m$, exactly as it was done while constructing an interpolation operator. Applying this procedure seems to result in a better convergence rate. Note also that the resulting coarse-level matrix will be weakly diagonally dominant (sum of each row's entries is zero).

3. There is no clear advantage in using the dynamic threshold for the subsonic flow case due to the increasing grid complexity, despite the slightly better convergence rate. A fixed threshold with $\varepsilon = 0.25$ works well for $M_\infty < 1$. However, in the supersonic flow regime, where the problem is of the hyperbolic type, applying the dynamic threshold improves the coarsening process and the convergence rate substantially, although the grid complexity deteriorates slightly. It is important to mention that the situation is even more clear-cut when the same operator is applied but with Dirichlet or Neumann boundary conditions. In this case, the dynamic threshold makes even a greater difference.

4. Both pointwise relaxation methods, Jacobi and SGS, are convergent, but the latter gives a faster convergence for the problems we have tested. In the case of supersonic flow, when the problem does not constitute an M-matrix, SGS clearly has a significant advantage over the Jacobi method.

5. In the sonic flow regime, $M_\infty = 1$ and when the flow is grid aligned, during the coarsening process by applying the direct approximation process (see Section 7), small entries are introduced in the matrix A. The entries in the ith equation, $i \in C^m$ (see (103)) are simply set to zero if the condition $\left|a_{ij}^m / a_{ii}^m\right| \leq \varepsilon_c$ is satisfied. Our first attempt to deal with this issue was as follows: since the corresponding points can no longer serve as coarse-grid points, they are ignored in the coarse-level; in other words, the corresponding equation is eliminated. In practice, these points are no longer considered as C-points thus becoming F-points. This results in poor interpolation since some of the strong connections are no longer considered as such. As a result, this procedure leads to a poor convergence factor. In order to resolve this difficulty properly it is necessary to understand its origin. Since the sonic case results in an operator with strong connections orthogonal to the flow direction (along the y-axis), the best coarsest level we can hope for has to include a certain minimal number of coarse-grid points along each of the grid's columns. Insisting on continuing the coarsening process beyond this level gives rise to the above difficulty. The main idea is that the coarsening process should be terminated when the problem can be solved efficiently

with just a few relaxations, although the number of remaining points on the coarse-level still seems relatively large.

6. In the sonic case, the damped SGS method had to be applied in the coarse-levels, since the regular SGS failed to converge. The value of the under relaxation parameter used $\omega = 0.7$. This, together with a timely termination of the coarsening process (see above), resulted in a robust algorithm with a bounded convergence factor ($C_f = 0.07$) for the sonic case, independent of the problem size. It was also verified (for the sake of constructing a uniform algorithm, suitable for the variety of cases) that the damped SGS performs well (as with $\omega = 1$) for the entire range of the flow regimes.

7. In the sonic flow regime $M_\infty = 1$; the interpolation operator constructed by transposing the restriction operator gave the best results. However, its performance is inferior in some other cases. Therefore, it cannot be a part of the overall "general purpose" algorithm.

9 The nonlinear transonic small disturbance (TSD) equation

So far we have considered linear problems. We will now discuss how AMG can be used to solve nonlinear problems. The framework for applying AMG to such problems can be as follows: The full problem on the finest grid is addressed directly by some kind of nonlinear relaxation as a smoother. Then the AMG cycle is applied to the linearized residual problem. The solution correction can be incorporated into the full nonlinear problem at the end of each multigrid cycle, and so on, until a converged solution is reached. The main objective of the reported research direction is to develop an AMG solver for the full potential equation in general geometries relying on body-fitted grids methodology. Here we illustrate the basic capabilities of the new methodology by applying it to a simpler (also nonlinear) case - transonic small disturbances equation (TSD). More detailed information about the derivation of the TSD equation can be found in references [12, 86, 15]. We shall present here just an overview.

The TSD equation can predict the transonic flow field about thin airfoils or through slightly convergent/divergent nozzles. It is derived by simplifying the full potential equation. The assumptions we make here are:

1. The airfoil is thin and its boundary's slope, $\frac{dy}{dx}$, is very small.

2. The local flow velocity components are close to the free-stream values.

It is assumed that the potential takes the form $\phi = V_\infty x + \varphi$ where V_∞ is the free-stream velocity and φ is the perturbation potential. Thus, the partial derivatives with respect to x and y become $\phi_x = V_\infty + \varphi_x$ and $\phi_y = \varphi_y$. The idea behind this assumption is that if we assume that the perturbation quantities are small, the second order terms (product of two small quantities) can be neglected. One of the corollaries from the small disturbance assumptions, is then the following

$$\frac{\varphi_x}{V_\infty} \ll \frac{\varphi_y}{V_\infty} \ll 1. \tag{111}$$

We start from the full potential equation in the quasi-linear form, and substitute: $\phi_{xx} = \varphi_{xx}$, $\phi_{yy} = \varphi_{yy}$, $\phi_{xy} = \varphi_{xy}$. The second term in (65) can be viewed as

$$2uv\varphi_{xy} = 2\left(V_\infty + \varphi_x\right)\varphi_y\varphi_{xy} = V_\infty^2\left(1 + \frac{\varphi_x}{V_\infty}\right)\left(\frac{\varphi_y^2}{V_\infty^2}\right)_x \cong 0, \tag{112}$$

since $\frac{\varphi_y^2}{V_\infty^2} \ll 1$. Consider the coefficient $(a^2 - u^2)$. Since the speed of sound and the flow velocity are related by the energy equation, (47), this coefficient can be approximated as

$$\begin{aligned} a^2 - u^2 &= a_\infty^2 + \frac{\gamma-1}{2}\left[V_\infty^2 - u^2 - v^2\right] - u^2 \\ &= a_\infty^2 + \frac{\gamma-1}{2}V_\infty^2\left[1 - \left(1 + \frac{\varphi_x}{V_\infty}\right)^2 - \left(\frac{\varphi_y}{V_\infty}\right)^2\right] - V_\infty^2\left(1 + \frac{\varphi_x}{V_\infty}\right)^2 \\ &= a_\infty^2 - V_\infty^2 - (\gamma+1)\,V_\infty^2\left(\frac{\varphi_x}{V_\infty}\right) \\ &= a_\infty^2\left[1 - M_\infty^2 - (\gamma+1)\,M_\infty^2\left(\frac{\varphi_x}{V_\infty}\right)\right], \end{aligned} \tag{113}$$

where M_∞ is the free stream Mach number and γ is the ratio of specific heats. The second powers of the disturbance velocity components are neglected as small. The coefficient of the third term in (47) can be approximated in a similar manner:

$$\begin{aligned} a^2 - v^2 &= a_\infty^2 + \frac{\gamma-1}{2}\left[V_\infty^2 - u^2 - v^2\right] - v^2 \\ &= a_\infty^2 + \frac{\gamma-1}{2}V_\infty^2\left[1 - \left(1 + \frac{\varphi_x}{V_\infty}\right)^2 - \left(\frac{\varphi_y}{V_\infty}\right)^2\right] - V_\infty^2\left(\frac{\varphi_y}{V_\infty}\right)^2 \cong a_\infty^2. \end{aligned} \tag{114}$$

Substituting these approximations we get the transonic small disturbance (TSD) equation:

$$\left[1 - M_\infty^2 - (\gamma+1)M_\infty^2\varphi_x\right]\varphi_{xx} + \varphi_{yy} = 0. \tag{115}$$

The TSD equation is nonlinear (the coefficient in front of φ_{xx} depends on the flow velocity). It is this nonlinearity that makes formation of shock waves possible. The mathematical difficulties of the problem are associated primarily with the mixed hyperbolic and elliptic type of the equations and the presence of discontinuities. The computational method should be capable of addressing these difficulties including the prediction of the shock formation.

The TSD equation may be elliptic or hyperbolic. In order to examine this behavior let is write (115) as:

$$C\varphi_{xx} + \varphi_{yy} = 0, \tag{116}$$

where,

$$C = 1 - M_\infty^2 - (\gamma+1)M_\infty^2\varphi_x. \tag{117}$$

The character of this equation may be determined by finding the roots of this equation, obtained from the theory of characteristics:

$$\frac{dy}{dx} = \pm\frac{1}{\sqrt{-C}}. \tag{118}$$

When $C > 0$ no real characteristics exist and the equation is elliptic. When $C \leq 0$ two characteristics exist with slopes that are equal in magnitude and opposite in sign, and will be symmetric about the x-axis. In this case (115) becomes hyperbolic.

Small disturbance approximation for the surface pressure coefficient C_p is as follows:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{\frac{p}{p_\infty} - 1}{\frac{1}{2}\frac{\rho_\infty}{p_\infty}u_\infty^2} = \frac{\left(\frac{\rho}{\rho_\infty}\right)^\gamma - 1}{\frac{1}{2}\gamma M_\infty^2}, \tag{119}$$

where C_p is the difference between local static pressure and free stream static pressure, nondimensionalized by the free-stream dynamic pressure. The term $\left(\frac{\rho}{\rho_\infty}\right)^\gamma$ can be replaced by (61), and we get:

$$\begin{aligned} C_p &= \frac{\left[1 + \frac{\gamma-1}{2}M_\infty^2\left(1 - \frac{u^2+v^2}{V_\infty^2}\right)\right]^{\frac{\gamma}{\gamma-1}} - 1}{\frac{1}{2}\gamma M_\infty^2} \\ &= \frac{\left[1 + \frac{\gamma-1}{2}M_\infty^2\left(1 - \frac{(V_\infty+\varphi_x)^2+\varphi_y^2}{V_\infty^2}\right)\right]^{\frac{\gamma}{\gamma-1}} - 1}{\frac{1}{2}\gamma M_\infty^2} \\ &\quad \frac{\left[1 + \frac{\gamma-1}{2}M_\infty^2\left(-2\frac{\varphi_x}{V_\infty}\right)\right]^{\frac{\gamma}{\gamma-1}} - 1}{\frac{1}{2}\gamma M_\infty^2}. \end{aligned} \tag{120}$$

Again, this approximation is obtained by neglecting the second order terms. This pressure coefficient is an immediate practical result from any aerodynamic computation, since it allows us to calculate drag and lift forces acting on a body. It is also a great tool to evaluate a solver.

9.1 The TSD discretization

In the present flow problem the finite differences are used for simplicity, and also because Murman and Cole's [20] original work used this type of discretization. Since the perturbation potential values are located at the cell's center, the domain boundary is located adjacent to the cell's centers. If the flow is subsonic and the equation is elliptic, every term in (115) will be computed, using the central finite difference formulas:

$$\begin{aligned} \varphi_x &= \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{\triangle x_{i+1,j} + \triangle x_{i,j}}, \\ \varphi_{xx} &= \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\left(\triangle x_{i+1,j} + \triangle x_{i,j}\right)^2}, \\ \varphi_{yy} &= \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\left(\triangle y_{i,j+1} + \triangle y_{i,j}\right)^2}, \end{aligned} \tag{121}$$

where $\triangle x$ and $\triangle y$ are the grid spacings in x and y directions, respectively. If the flow is supersonic, then the φ_x and φ_{xx} terms must be shifted in the direction of the flow, while the term φ_{yy} (derivative in the normal to the flow direction) is evaluated using central differences as written in (121). To summarize, the derivatives are approximated as follows:

$$\begin{aligned} \varphi_x &= \frac{\varphi_{i,j} - \varphi_{i-1,j}}{\triangle x_{i,j}}, \\ \varphi_{xx} &= \frac{\varphi_{i,j} - 2\varphi_{i-1,j} + \varphi_{i-2,j}}{\left(\triangle x_{i,j} + \triangle x_{i-1,j}\right)^2}, \\ \varphi_{yy} &= \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\left(\triangle y_{i,j+1} + \triangle y_{i,j}\right)^2}. \end{aligned} \tag{122}$$

We shall emphasize again that the discretization is determined by the sign of the quantity C given by (117). If $C > 0$, the flow is subsonic and a central differencing is used, while the supersonic case is characterized by $C < 0$ and an upwind approximation is used.

As an initial condition to start the iterative process we set the potential perturbation φ at all interior points to zero. That is,

$$\varphi_{i,j} = 0. \tag{123}$$

Once the problem is solved and the values of φ are known at all the points, we can evaluate other quantities such as disturbance velocities, Mach number, and surface coefficient pressure.

9.2 Overview of the solution procedure

The basic steps in the iterative solution of the TSD equation while applying the AMG method are as follows:

1. Construct the grid.
2. Compute the grid spacings $\triangle x$ and $\triangle y$ and store them in arrays.
3. Initialize the potential by $\varphi = 0$ in all the computational domain.
4. Construct the coefficient matrices A and $\tilde{A}$ by a linearization process. Then the product matrix $A^* = A\tilde{A}$ is evaluated. This matrix will be used in the coarsening process, and the coarse-level correction process.
5. Next is the setup phase, which includes the coarse-grid selection and the design of appropriate interpolation and restriction operators. Based on the matrix A^* the coarse point selection algorithm proceeds in two passes. We first make an initial division of the grid points by choosing a preliminary partition into C- and F-points. Once the initial assignment have been made, we make a second pass in order to satisfy the coarsening requirements as described in Section 4. In this way the coarse-level and the restriction operator are constructed. Next the interpolation operator is constructed based on the algorithm described in Section 4.

6. At this point the solving phase begins. The V-cycle algorithm is executed with two SGS relaxations on each coarse-level, for both the downward and upward directions. In the coarsest level the problem is usually solved after 5 to 15 relaxations and then the correction is interpolated and added to the fine-level. It is important to note that on the fine-level only, direct SGS relaxations were done (not with the matrix A^*). This reason for this is that we could have the ability to apply local extra relaxation sweeps in cases when the nonlinearity is dominant, for example, near shocks waves or stagnation points.

7. The goal is to reduce the L_2-norm of the residual below 10^{-10}, with a convergence rate of less than an order of magnitude per V-cycle. In the case of $M < 0.5$ the set up phase is performed only one time, at the beginning of the algorithm, and no extra updates of the coefficient matrices A, $\tilde{A}$, and A^* are needed. When the flow is transonic, the nonlinearity is dominant and each V-cycle is followed by a new setup phase in order to achieve the desired convergence rate. This is done since the nonlinearity is solved on the fine-level only. Since the matrices considered are constructed of nonlinear coefficients, they must be updated. Performing the setup phase is expensive in terms of storage, computation, and time, but since the nonlinearity is solved only on the fine-level, it is more than necessary in order to achieve convergence. The setup phase can be performed after 1-2 V-cycles and it is problem dependent. In the TSD equation the setup phase is performed 6 times with two V-cycles between each update, resulting in 12 V-cycles.

8. After the residual reaches a sufficient level, the post processing starts. Once the potential perturbation φ in all the points is known we can compute the velocity components u and v as follows: $u = \varphi_x + V_\infty$, $v = \varphi_y$. Then the local Mach number M_{ij}, and the pressure coefficient C_p are computed.

9.3 Subsonic flow in a channel with a bump

The cases considered here are subsonic and transonic flow in a channel with a bump. A uniform orthogonal H-grid used (96 points in the x-direction and 48 points in the y-direction) is presented in Figure 16. The perturbation potential values are located at the centers of the cells. Since the equation is discretized with finite difference method, the cells have no meaning in the numerical process.

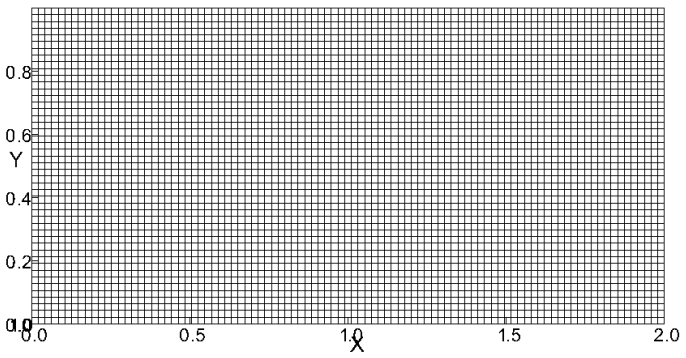


Figure 16: The mesh used for solving the TSD equation.

The influence of the bump on the flow is imposed through the boundary conditions; the velocity component normal to the "surface" $V_\perp$ is approximated as follows:

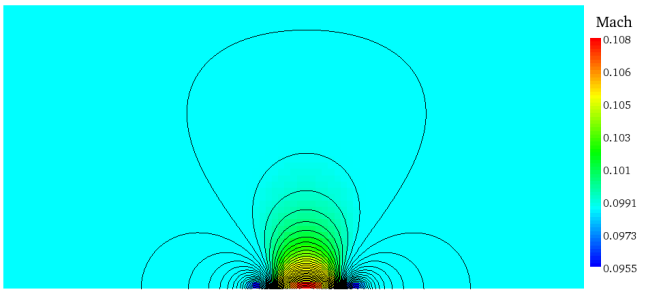
$$V_\perp = f' \cdot V_\infty, \tag{124}$$

where in our case

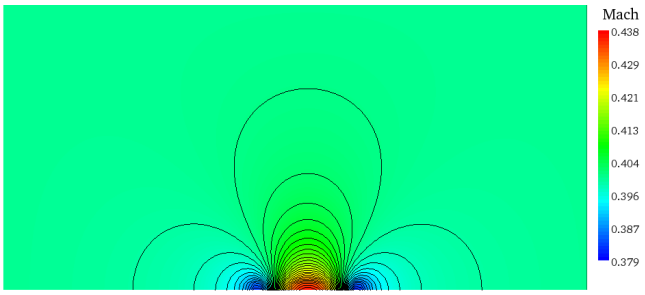
$$f' = -\frac{(x-48)}{\sqrt{R^2-(x-48)^2}}, \tag{125}$$

and it represents a surface slope of a circular arc of $\pi/15$ and radius 192. The bump is located at the lower wall of the channel at $44 \leq x \leq 54$.

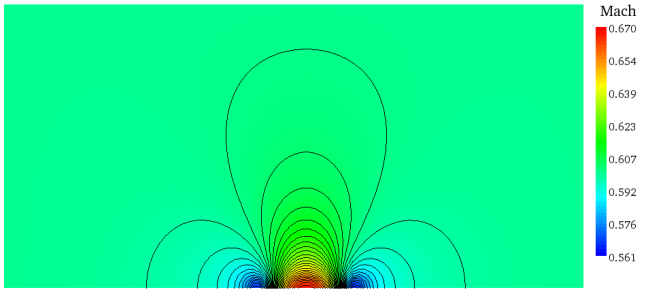
Attention is now turned to the boundary conditions. Since we are solving here the equation for the potential perturbation that is caused by the existence of the bump at the center of the lower wall, a good assumption is that the perturbation is zero at the farfield boundaries. Therefore in the inlet and outlet boundaries a homogeneous Neumann condition was applied, while in the upper wall a Dirichlet boundary condition was applied. The difference at the points next to the boundaries were evaluated using ghost points. The nonlinear SGS relaxation was implemented as a smoother on the finest level. The Mach number isolines and color maps are shown in Figure 17 for four cases of inlet velocity (Mach number) varying from 0.1 to 0.9. Fifty isolines have been drawn on the color map for each case, ranging from the lowest velocity to the highest velocity value. It is clear that increasing the Mach number makes the compressibility effects more dominant and for an incident velocity of $M_\infty = 0.9$ a supersonic region is reached on the bump, which is terminated by a shock.



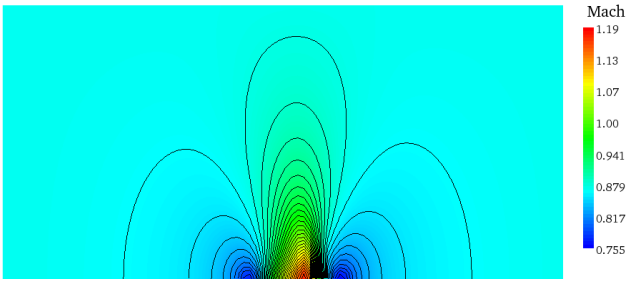
(a)



(b)



(c)



(d)

Figure 17: Transonic flow over a circular bump; free stream $M_\infty = 0.9$, grid 96×48 points.

The pressure coefficient distribution on the lower surface is plotted in Figure 18. Note that the pressure on most of the surface is less than $p_\infty$and the minimum value of C_p on the surface is -3 at $\theta = \pi/2$.

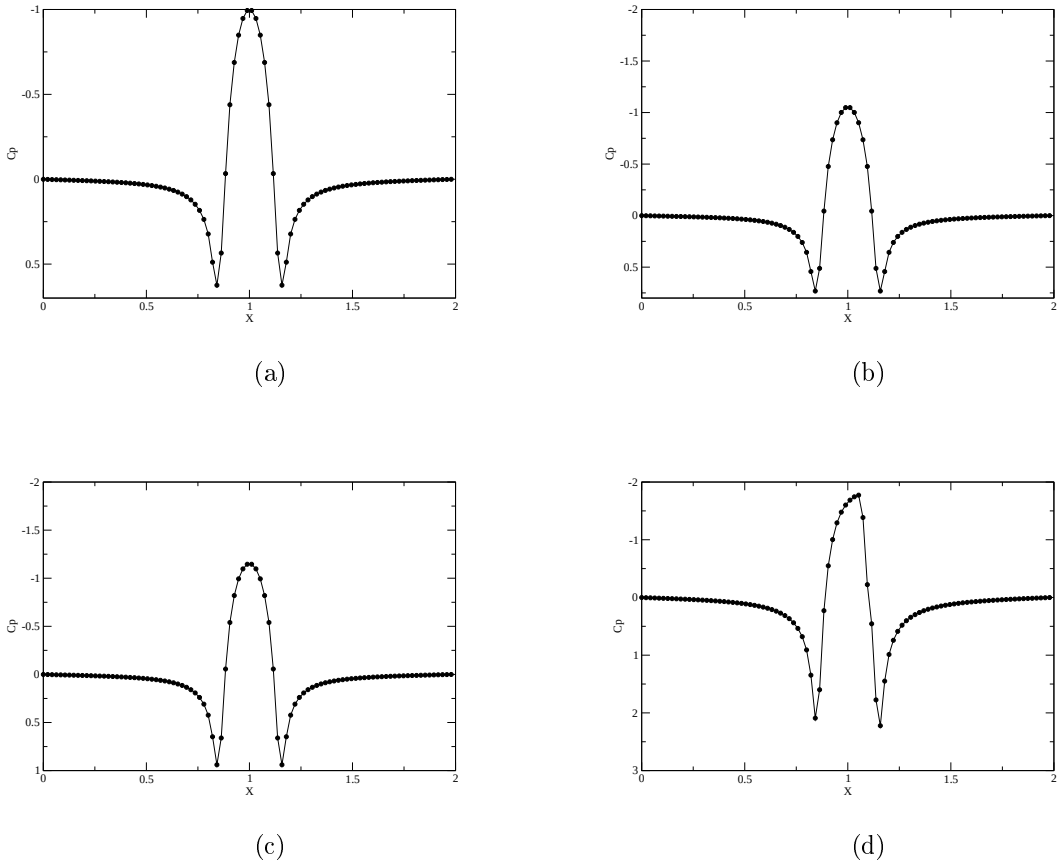


Figure 18: Surface pressure coefficient C_p along the bottom wall with an incident Mach number ranging from $M_\infty = 0.1$ to $M_\infty = 0.87$, using the (96×48) mesh size.

AMG performance

The first coarse-level for each case described above is sketched in Figure 19. When of Mach number $M_\infty < 0.3$ the compressibility effects are not yet significant and the operator is nearly isotropic, the points that construct the coarse-level are distributed uniformly, as can be seen in Figure 19. As the free-stream velocity is increased to nearly the sonic case, the problem's attribute is strong dependence in the y-direction, and little or no dependence in the x-direction. This is clearly illustrated by Figure 19(a). The algorithm generates a semicoarsened grid that is coarsened only in the y-direction, in the direction of the strong dependence. This is precisely the coarse-grid that one would use to construct an efficient geometric multigrid method. Since smooth error varies slowly in the direction of the strong dependence, the interpolation can be performed accurately in that direction and the coarsening algorithm automatically exhibits this critical property.

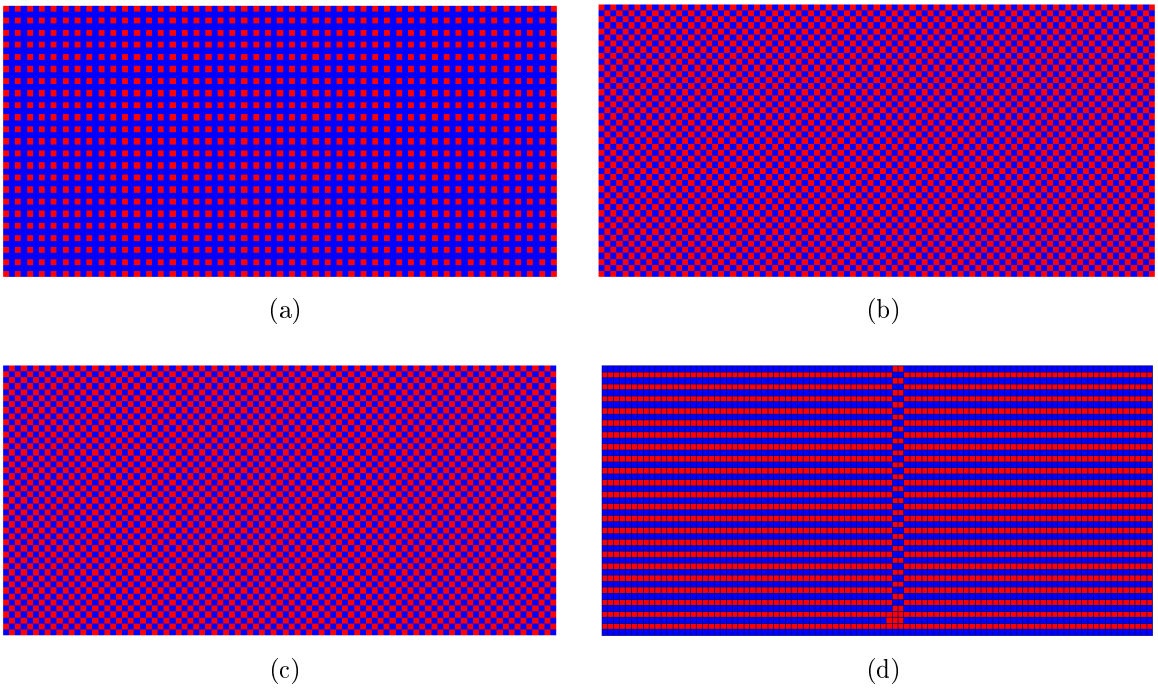


Figure 19: The fine and first coarse-level produced for the TSD equation in several flow conditions: a) $M_\infty = 0.1$ b) $M_\infty = 0.4$ c) $M_\infty = 0.6$ d) $M_\infty = 0.87$. The blue cells correspond to F-points and the red cells correspond to the C-points.

The L_2-norm of the residual is shown in Table 4 for each V-cycle, for four different cases of inlet velocity. Note that for the first three cases and for all the grid sizes, the convergence factors are bounded well below 0.1. The residual norm decreases by a nearly constant factor with each V-cycle. This continues until it levels off after about 10 V-cycles near 10^{-13}, where round-off error is on the order of the residual norm itself. The fast convergence is mainly due to the particularly simple geometrical situation, which benefits the interpolation process. We will see later that this advantage gets lost in more complex geometric situations or for more complicated problems (FPE in conservation form). To illustrate the residual reduction graphically, Figure 20 presents the convergence history of the residual (L_2-norm) versus the iterations number, for the four cases described above.

The convergence properties in the case of $M_\infty = 0.87$ deteriorate slightly due to the extreme anisotropy of the problem and, perhaps even more, due to the presence of a shock wave. The coefficient matrix used for evaluating the AMG operators is constructed by a linearization of the problem. The main reason for the performance degradation seems to be that the coefficients, especially in the transonic flow regime, depend rather strongly on the solution. A useful technique to overcome this problem is to apply local smoothing sweeps (about 2-3 extra local relaxation sweeps) near the shock wave, immediately after distributing the correction to the points on the fine-level. Often, the extra relaxations overcome any difficulty near the shock waves and the overall results approached the desired convergence factor of 0.1. The additional work does not seriously affect the overall complexity and convergence properties since the number of points that forms the

shock wave is usually very small in comparison with the number of interior points (set of measure zero). The performance of the AMG with 2-3 extra relaxations at the end of each cycle are presented in Table 5.

Table 4: The results of AMG V-cycles applied to the flow through a channel with a bump. The second norm of the residual $\|R^m\|_2$ and the convergence factor C_f are presented. The mesh size is 96×48.

	$M_\infty = 0.1$		$M_\infty = 0.4$		$M_\infty = 0.6$		$M_\infty = 0.9$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	2.518e-00	–	10.306e-00	–	15.971e-00	–	0.1205e-00	–
1	6.254e-02	0.02	0.184e-00	0.02	0.480e-00	0.04	0.174e-00	1
2	1.517e-03	0.03	3.203e-03	0.02	1.973e-02	0.05	3.599e-02	0.21
3	3.932e-05	0.03	6.104e-05	0.02	9.200e-04	0.05	6.452e-03	0.18
4	1.031e-06	0.03	1.182e-06	0.02	4.242e-05	0.05	8.664e-04	0.13
5	2.700e-08	0.03	2.283e-08	0.02	1.921e-06	0.04	1.164e-04	0.13
6	7.022e-010	0.03	4.371e-10	0.02	8.599e-08	0.04	1.550e-05	0.13
7	1.816e-11	0.03	8.295e-12	0.02	3.821e-09	0.04	2.034e-06	0.13
8	4.677e-13	0.03	1.542e-13	0.02	1.690e-10	0.04	2.636e-07	0.13
9	1.221e-14	0.03	1.192e-14	0.08	7.458e-12	0.04	3.380e-08	0.13
10	2.816e-15	0.23	1.183e-14	0.99	3.311e-13	0.04	4.290e-09	0.13
11	2.751e-15	0.98	1.176e-14	0.99	2.377e-14	0.07	5.394e-10	0.13
12	2.664e-15	0.97	1.156e-14	0.98	1.912e-14	0.80	6.731e-11	0.12
13	–	–	–	–	–	–	8.348e-12	0.12
14	–	–	–	–	–	–	1.035e-12	0.12

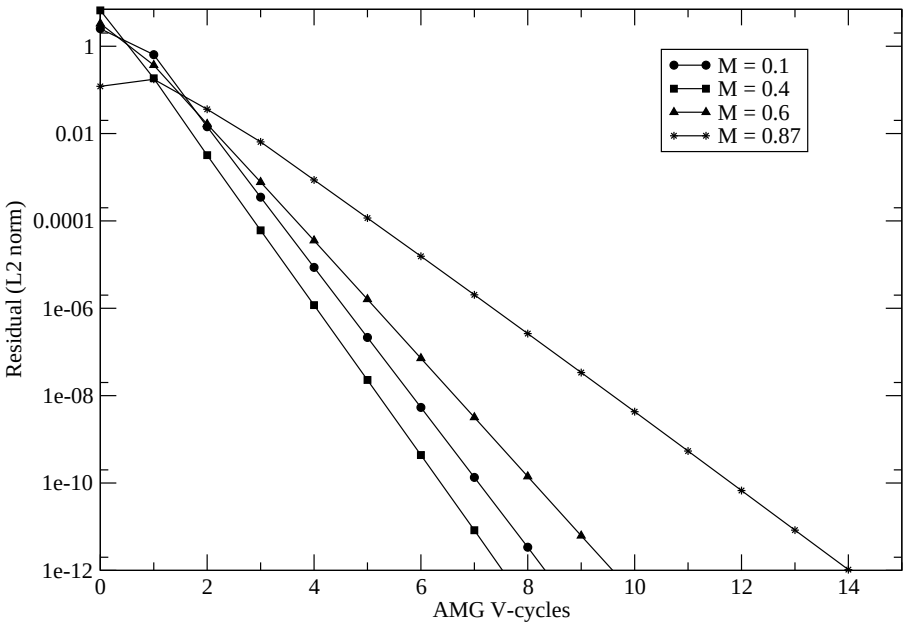


Figure 20: Convergence history of the residual.

Table 5: The table shows the results of AMG V-cycles applied to the flow through a channel with a bump. The second norm of the residual $\|R^m\|_2$ after each V-cycle, convergence factor C_f, are presented for $M_\infty = 0.87$. The mesh size is 96×48.

$M_\infty = 0.87$		
V-cycle	$\|R^m\|_2$	C_f
0	0.1205e-00	–
1	4.062e-02	0.33
2	6.972e-03	0.17
3	6.186e-04	0.09
4	7.096e-05	0.11
5	7.698e-06	0.10
6	7.910e-07	0.10
7	8.123e-08	0.11
8	8.793e-09	0.12
9	1.011e-09	0.12
10	1.207e-10	0.12
11	1.464e-11	0.12
12	1.792e-12	0.12
13	2.202e-13	0.12
14	6.053e-14	0.27

Table 6 presents the grid and operator complexity for four test cases with different Mach numbers. Summing the number of rows of all the coarse-level operators and dividing by the number of rows on the fine-level shows the grid complexity. For example, in the case of $M_\infty = 0.1$ the grid complexity is 1.497. Thus, we know that the total storage of the vector of unknowns φ, and the right side require 1.497 times the space required for the fine-grid quantities. As a comparison, the geometric approach has a grid complexity of about $\frac{4}{3}$. The difference can be explained as follows: despite the fact that the number of points on the first coarse-level (1152) is exactly one-fourth that of the fine-level (4608), a thing that in the geometric approach is obtained by full coarsening, the other coarse-levels has fewer than half the number of points as the next finer level. In the case of $M_\infty = 0.4$ the first coarse-level produced by the AMG is a red-black (coarsened by a factor of ~2).

Summing the number of nonzeros in all the operators and dividing by the number of nonzeros in the fine-level shows that the operator complexity is greater than 2 for the four cases of Mach number. This parameter indicates how much storage is needed. The operator complexity also reflects the cost of one relaxation sweep on any level, so consider the case of $M_\infty = 0.1$, a V(2,2) cycle of AMG costs about 9.16 WUs (2.29WU on the descent and the ascent). The operator complexity is slightly increased with the Mach number and affects the number of operations required. It can be observed that the operator complexity increases with the Mach number. Two factors affecting this are the average stencil size and the coarsening process. The average stencil size is the average number of nonzero coefficients per row. For simplicity, let us take a look at the fine-level. For the low Mach number, the stencil size of the matrix A^* is quite small, since $\tilde{A}$ is nearly a unity matrix. As the Mach number increases so does the average stencil of A^* due to the structure of $\tilde{A}$. It is possible to get very large stencil sizes on coarser levels. Large stencil size can lead to large operator complexity since various processes such as coarsening, interpolation, and relaxation require that neighbors of neighbors are visited, which results in a growth in the number of operations per cycle. The second reason for the increased operator complexity is the relatively large number of points on the fine-levels, which is partially a result of the second pass process, while F-points are replaced by C-points in order to satisfy the interpolation requirements.

The convergence factor, grid complexity, and stencil size were considered when the coarsening and interpolation procedures were defined, since they affect each other. Increasing complexities can improve convergence, while decreasing grid and operator complexities lead to a degradation in convergence. Maintaining a convergence factor around an order of magnitude, while keeping the complexities as small as possible is the first priority in this work. Several remarks concerning the trade-off between convergence rate and memory requirements are detailed in Section 12.

Table 6: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

complexities	$M_\infty = 0.1$	$M_\infty = 0.4$	$M_\infty = 0.6$	$M_\infty = 0.87$
C_Ω	1.497	1.971	1.920	1.942
C_L	2.294	3.207	2.873	3.208

This problem was solved on two more grids with different resolutions of 48×24, and 24×12. To illustrate the AMG performance with respect to the problem size, the extreme case with an ambient Mach number of $M_\infty = 0.87$ is presented in Table 7. The results were obtained by repeating the setup phase 6 times while applying two V-cycles between each update. This results in a total of 12 V-cycles until the residual has decreased to the desired level of $\tilde{}10^{-10}$. It can be seen clearly that the algorithm is scalable and does not depend on the problem size (the convergence factor is nearly constant for all the grids considered).

Table 7: The results of AMG V-cycles applied to the flow through a channel with a bump. The second norm of the residual after each V-cycle, the convergence factor, grid complexity, and operator complexity are presented for two grid sizes.

$M_\infty = 0.87$				
grid size $\longrightarrow$	48×24		24×12	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	5.781e-00	–	2.666e-00	–
1	0.644e-00	0.5	0.310e-00	0.30
2	0.319e-00	0.10	9.347e-02	0.20
3	3.214e-02	0.16	1.861e-02	0.12
4	5.018e-03	0.18	2.256e-03	0.11
5	8.796e-04	0.10	2.579e-04	0.10
6	8.813e-05	0.09	2.630e-05	0.10
7	7.923e-06	0.09	2.743e-06	0.10
8	6.999e-07	0.09	2.869e-07	0.10
9	6.278e-08	0.09	3.012e-08	0.10
10	5.756e-09	0.09	3.165e-09	0.11
11	5.371e-10	0.09	3.328e-10	0.11
12	5.067e-11	0.09	3.498e-11	0.11

10 Structured finite volume scheme

In this section, the methodology developed to solve the FPE in the conservation form on body-fitted structured grid will be covered.

10.1 Finite control volume

The derivation of the principal equations of fluid dynamics is based on the fact that the dynamic behavior of a fluid is determined by the conservation laws (mass, momentum, and energy). The conservation of a flow quantity means that its total variation inside an arbitrary volume can be expressed as the net effect of the amount of the quantity across the boundary and of external forces acting on the volume. The quantity crossing the boundary is called flux. In order to develop a mathematical description of the control volume let us consider a general 2D flow field as represented in Figure 7. Since the formulation is applied for 2D meshes, the cell volume is simply the area. In a similar way, the face area is the length of the face. A finite region of the flow, bounded by the closed surface $d\Omega$ and fixed in space, defines the control volume Ω. A surface element is assigned as dS, and its associated unit normal vector is $\overrightarrow{n}$, and it is defined as $\hat{n} = n_x\hat{i} + n_y\hat{j}$. It is defined as positive when pointing outward from the control volume surface. The conservation law to a scalar quantity per unit volume U is written as follows:

$$\frac{d}{dt}\int (U \cdot d\Omega)\,. \tag{126}$$

The variation of U in time is equal to the amount of the quantity U entering the control volume through the boundary with the velocity V

$$-\int \left(U\left(\overrightarrow{V}\cdot\overrightarrow{n}\right)ds\right). \tag{127}$$

Due to the diffusive flux that is expressed by the Fick's law (relating the diffusive flux to the concentration field)

$$\oint_{\partial\Omega} \kappa\rho\left[\nabla\left(\frac{U}{\rho}\right)\cdot\overrightarrow{n}\right]ds, \tag{128}$$

Whereκ is the thermal diffusivity coefficient. The volume Q_v and surface sources Q_s are expressed as

$$\int_{\partial\Omega} Q_v d\Omega + \oint_{\partial\Omega}\left(\vec{Q}_s\cdot\overrightarrow{n}\right)ds. \tag{129}$$

After summing the above contributions we obtain the general form of the conservation law for the scalar quantity U

$$\frac{\partial}{\partial t}\int_{\Omega} U d\Omega + \oint_{\partial\Omega}\left[U\left(\overrightarrow{v}\cdot\overrightarrow{n}\right) - \kappa\rho\left(\nabla\left(\frac{U}{\rho}\right)\cdot\overrightarrow{n}\right)\right]ds = \int_{\Omega}\vec{Q}_v d\Omega + \oint_{\partial\Omega}\left(\vec{Q}_s\cdot\overrightarrow{n}\right)ds. \tag{130}$$

If the conserved quantity is a vector, the convective and the diffusive flux would become tensors, also the volume and surface sources would change into a tensors. The conservation law for a general vector quantity U is

$$\frac{\partial}{\partial t}\int_{\Omega} U d\Omega + \oint_{\partial\Omega}\left[\left(\overline{\overline{F}}_c - \overline{\overline{F}}_D\right)\cdot\overrightarrow{n}\right]ds = \int_{\Omega}\vec{Q}_v d\Omega + \oint_{\partial\Omega}\left(\overline{\overline{Q}}_s\cdot\overrightarrow{n}\right)ds, \tag{131}$$

where $\overline{\overline{F}}_c$ is the convective flux tensor and $\overline{\overline{F}}_D$ is the diffusive flux tensor. If there are no volume sources, U depends solely on the flux across the boundary $d\Omega$.

10.2 The structured finite volume scheme

The structured finite volume scheme is based on the conservation laws, which are expressed by the Navier–Stokes, Euler, or the FPE [87, 88]. The physical space is divided into a number of grid cells — quadrilaterals in $2D$. The grid generation is done in such a way that: the domain is completely covered by the grid, there is no free space left between the grid-cells, and the grid-cells do not overlap each other (Some excellent books of grid generation methods can be found in [89, 90, 91, 92]). The resulting structured grid is described by the Cartesian coordinates x, y, and z (the corners of the grid cells). Based on the grid, the control volume is defined in order to evaluate the integrals of the convective fluxes as well as of the source term. Since a control volume does not change in time, we get $\frac{d}{dt}\int (w d\Omega) = \Omega \frac{dw}{dt}$. Eq. (126) becomes

$$\frac{\partial \overrightarrow{W}}{\partial t} = -\frac{1}{\Omega}\left[\oint_{\partial\Omega}\left(\vec{F}_c - \vec{F}_v\right) dS - \int_{\Omega} \overrightarrow{Q}\, d\Omega\right]. \tag{132}$$

The flux is evaluated at the midpoint of the face. Considering a particular volume $\Omega_{i,j}$ we obtain

$$\frac{d\vec{W}_{i,j,k}}{dt} = -\frac{1}{\Omega_{i,j,k}}\left[\sum_{m=1}^{N_F}\left(\vec{F}_c - \vec{F}_v\right)_m \triangle S_m - \left(\overrightarrow{Q}\Omega\right)_{i,j,k}\right]. \tag{133}$$

The variable $\triangle S_m$ presents the area of face m. The term in the square brackets is simply the residual $R_{i,j}$.

10.3 Geometrical quantities of the control volume

Before the discretization process the area of the control volume $\triangle S_m$, unit normal vector n_m, and the coordinate systems involved must be calculated and stored. In this work a cell-centered scheme was preferred – the control volumes are identical with the grid cells and the flow variables are located at the cell's center. The center's coordinates of each cell, C_1 and C_2, are obtained by the arithmetic average of the four corner cells 1, 2, 3, and 4, as is sketched in Figure 21.

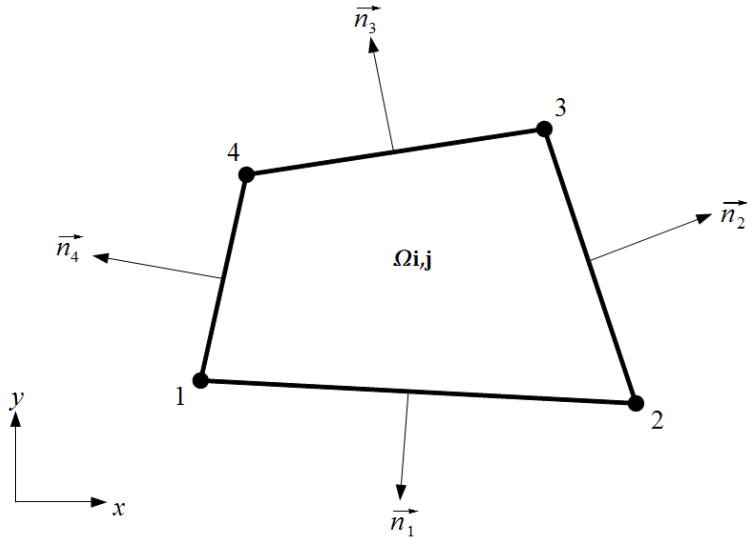


Figure 21: Control volume of a cell-centered scheme.

For instance, point $C\,(x, y)$ is defined by its coordinates,

$$C\,(x, y) = \left[\frac{(x1 + x2 + x3 + x4)}{4},\ \frac{(y1 + y2 + y3 + y4)}{4}\right]. \tag{134}$$

The area Ω is calculated as follows,

$$\Omega_{i,j} = \frac{V_{21}V_{41}}{2} + \frac{V_{32}V_{34}}{2}, \tag{135}$$

where the vectors V_{21}, V_{41}, V_{32}, and V_{34} are:

$$\begin{aligned} V_{21} &= (x_2,\, x_2) - (x_1,\, y_1)\,, \\ V_{41} &= (x_4\, y_4) - (x_1,\, y_1)\,, \\ V_{32} &= (x_3\, y_3) - (x_2,\, y_2)\,, \\ V_{34} &= (x_3\, y_3) - (x_4,\, y_4)\,. \end{aligned} \tag{136}$$

The faces of the control volume ("control area" since the 2D case is considered) are given by straight lines, therefore, the unit normal vector is constant along them. When we integrate the fluxes according to (132), we have to evaluate the product of the area of a face $\triangle S$ and the corresponding unit normal vector $\overrightarrow{n}$. The face vector $\overrightarrow{S}$ is defined as,

$$S_m = [S_x,\, S_y] = n_m \triangle S_m. \tag{137}$$

The face vectors of the control volume are given by the following relations:

$$
\begin{aligned}
S_1 &= \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \end{bmatrix}, \\
S_2 &= \begin{bmatrix} y_3 - y_2 \\ x_2 - x_3 \end{bmatrix}, \\
S_3 &= \begin{bmatrix} y_4 - y_3 \\ x_3 - x_4 \end{bmatrix}, \\
S_4 &= \begin{bmatrix} y_1 - y_4 \\ x_4 - x_1 \end{bmatrix}.
\end{aligned} \tag{138}
$$

The unit normal vector at the face m is obtained from (137) as $n_m = \frac{S_m}{\triangle S_m}$, while $\triangle S_m = |S_m| = \sqrt{S_x^2 + S_y^2}$.

In practice, all the unit normal vectors and the face lengths are computed and stored for each control volume $\Omega_{i,j}$.

10.4 Coordinate systems

In structured grids, the grid points in the physical space are mapped in a unique way onto a continuous set of two integers $(i,\ j)$ (one for each coordinate direction). The set of integers defines the computational space. Since the fluxes, according to Eq. (132), are computed in a direction normal to the face, a mechanism that allows the transformation of the covariant vectors (consider with the grid direction) to contravariant vectors (normal to the face) is needed. For this purpose three different coordinate systems are involved: Cartesian, covariant, and contravariant, as sketched in Figure 22.

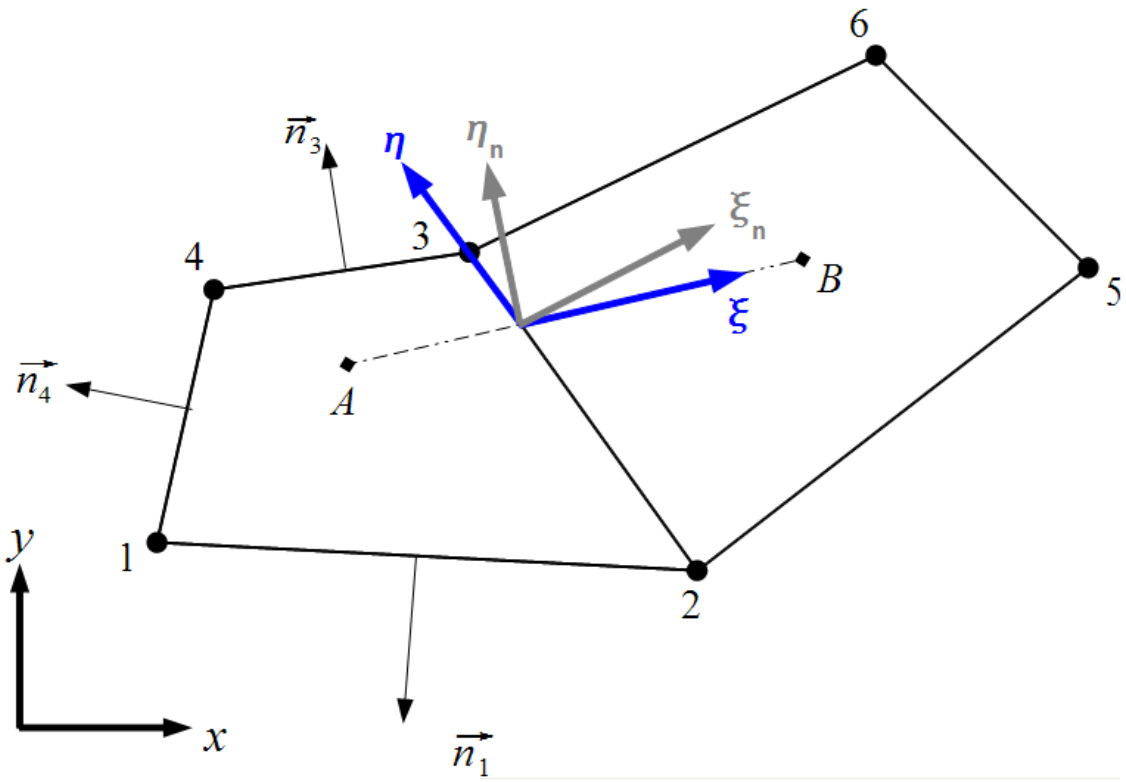


Figure 22: 2D cells and the associated three coordinate systems.

- Cartesian coordinate system $(x,\ y)$ – orthogonal coordinate system (the axes are mutually perpendicular) where the mesh grid is defined.

- Covariant coordinate system (ξ, η) – nonorthogonal coordinate system attached to the face of each element. Consider the face $2-3$ in Figure 22. In order to obtain the best scaling of the numerical values, the origin of the coordinate system is located on the intersection between the face $(2-3)$ and the line that connects the cell's centers C_1 and C_2. The ξ-axis connects two adjacent cells centers (C_1 and C_2), i.e.,

$$\xi = C_2\,(x, y) - C_1\,(x, y)\,, \tag{139}$$

where C_1 and C_2 are the cell's centers defined in a Cartesian coordinate system (x, y). The η-axis is aligned with the face $(2-3)$, and is defined as follows:

$$\eta = V_3\,(x_3,\ y_3) - V_2\,(x_2,\ y_2)\,, \tag{140}$$

where V_1 and V_2 are the grid's node (vertex) coordinates. Let us define the matrix M that transform the general vector from a Cartesian coordinate system (x, y) to the covariant coordinate system (ξ, η):

$$M = \begin{bmatrix} \hat{\xi}_x & \hat{\xi}_y \\ \hat{\eta}_x & \hat{\eta}_y \end{bmatrix}. \tag{141}$$

The first row in the matrix M includes unit vector components in the ξ-direction, $\left(\hat{\xi}_x, \hat{\xi}_y\right)$, and the second row includes a unit vector components in η-direction, $(\hat{\eta}_x, \hat{\eta}_y)$. The matrix M transform a general vector g from the Cartesian (orthogonal) coordinate system (x, y) to the covariant (non-orthogonal) coordinate system (ξ, η),

$$\begin{bmatrix} g_\xi \\ g_\eta \end{bmatrix} = M \cdot \begin{bmatrix} g_x \\ g_y \end{bmatrix}. \tag{142}$$

We need the covariant system in order to evaluate the equation derivatives. Since a body-fitted grid is applied, the grid is far from being orthogonal (especially adjacent to the body's surface) and hence the covariant vector does not coincide with the face normal direction. The convective fluxes are defined normal to the face element. For this purpose we need the contravariant coordinate system (ξ_n, η_n).

- Contravariant coordinate system (ξ_n, η_n) – a nonorthogonal coordinate system. Its origin coincides with the covariant coordinate system. The axis ξ_n is normal to the face element $(2-3)$ and the axis η_n is normal to the covariant ξ-axis, a general vector g in the Cartesian coordinate system can be transformed to the contravariant coordinate system by the following,

$$\begin{bmatrix} g_{\xi n} \\ g_{\eta n} \end{bmatrix} = M' \cdot \begin{bmatrix} g_x \\ g_y \end{bmatrix}, \tag{143}$$

where $M' = \left(M^T\right)^{-1}$. It is important to note that in the transformation from Cartesian to contravariant coordinate system, the magnitude of the general vector

$\overrightarrow{g}$ is reserved. This fact does not hold in the transformation from Cartesian to covariant coordinate system. In practice, the magnitude of a general vector g is obtained by,

$$|g| = \sqrt{V_\xi V_{\xi n} + V_\eta V_{\eta n}}. \tag{144}$$

The equation discretization takes place in each cell in the covariant coordinate system. Since the fluxes are computed normal to the face direction, a transformation mechanism from the covariant to the contravariant coordinate system is needed. In practice, the matrices M and M' are computed and stored for each face in the grid.

If the coordinate system is orthogonal, then the contravariant and covariant interpretations are identical (up to scale factors). This can be seen by imagining that we make the coordinate axes in Figure 22 perpendicular to each other. It is worth noting that orthogonal does not necessarily imply rectilinear. For example, in polar coordinates the axes are not straight lines, but they are orthogonal, because as we vary the angle we are always moving perpendicular to the local radial axis. See, for example, the mesh grid used to solve the flow around a circular cylinder in Section 13. When we consider systems of coordinates that are not mutually perpendicular, the contravariant and covariant forms differ. As an example see the problems presented in Section 13.

11 Discretization of the FPE in the conservation form

The purpose of the current section is to describe a procedure for constructing stable finite volume approximations to the conservation form of the full potential equation. The strategy of discretizing the FPE in the conservation form is based upon an idea similar to that of the rotated difference approach introduced by Jameson [27] and implemented initially in the quasi-linear form of the equation. However, this approach is not made directly. Instead it is accomplished indirectly by following the same rationale.

We review briefly our approach starting with the FPE in the quasi-linear form (72),

$$\nabla^2\phi - M^2\frac{\partial^2}{\partial s^2}\phi = 0. \tag{145}$$

Let us look at both terms (72) from a numerical standpoint. Note that when the Mach number is close to zero (incompressible flow) the second term can be neglected; thus we are left with $\nabla^2\phi$, which is discretized by a certain type central differencing, according to (77). As the Mach number increases the second term, which describes the second derivative in the streamwise direction, actually determines the "dynamics" of the flow. When the flow is subsonic, a central difference is used for the second derivative in the flow direction (ϕ_{ss}), while pointwise relaxation is applied directly in conjunction with the matrix A (or operator L). If the flow is supersonic, we apply relaxation directly with the product of downwind and upwind operators $L\tilde{L}$ (or matrix $A\tilde{A}$). We would like to apply the same rationale to the discretization of the FPE in the conservation form, while the advantages

that were obtained in the quasi-linear case discretization, would be implemented. The relation between the discretization approach applied in the quasi-linear form and the FPE in the conservation form will now be covered. Expanding the FPE and rearranging terms, Eq. (145) can be reformulated as,

$$\rho\nabla^2\phi + \left(\phi_x\frac{\partial}{\partial x} + \phi_y\frac{\partial}{\partial y}\right)\rho = 0, \tag{146}$$

where the density ρ is given in Eq. (61). Note that Eq. (145) and (146) have a similar structure. The density parameter ρ plays two roles. In the first term it serves as a constant. In the second term it serves as an unknown variable. As one can see, the second terms in both of the above equations are identical. Therefore, the same rationale applied in the quasi-linear case can be applied to the conservation form. The description of the discretization technique is as follows: For the first term $\nabla^2\phi$ the fluxes are computed by a central discretization independent of the flow direction and speed. The dynamics of the flow is reflected in the second term which is discretized in such a way that the result is a "wide" approximation in the streamwise direction.

The criterion for selecting the best discretization consists mainly of how accurate the discretization is in computing the gradients and how generally applicable the algorithm can be. An initial test to verify the discretization's accuracy is to check that the method can reproduce a free-stream velocity applied to an arbitrary mesh. If not, then the discretization will not be acceptable. The second criterion deals with how general the discretization is. It is desired to attain solutions on highly stretched irregular structured grids, and flow in various speeds and directions. The divergence of the fluxes, must be computed according to Eq. (132). The flux through a given cell's face is a product of the velocity vector and the density, which are both functions of the potential ϕ. The type of the discretization approach, central fluxes, or upwind fluxes, is determined by the local Mach number across the cell's face. So, the first obstacle in forming the flux is therefore calculating the velocity vector at each cell's face.

11.1 Velocity components

From Eq. (49), the velocity field can be found by calculating the gradients of ϕ in the x, y, and z directions. For an orthogonal structured grid, this calculation is straight forward due to the Cartesian ordering of the cells. For a nonorthogonal grid, it is less clear how to formulate and compute the gradients of ϕ. Before obtaining the divergence, however, one needs to define the physical components of the velocity vector for each face in the grid. The velocity components are derived in the covariant coordinate system (ξ, η). Consider for example the face at the half node $(i - 1/2, j)$ as sketched in Figure 23. The velocity vector of the flow through this face has two covariant components as follows:

$$V_{cov} = V_\xi\hat{\xi} + V_\eta\hat{\eta}, \tag{147}$$

where V_ξ is approximated by a "narrow" derivative in ξ-direction and V_η is derived by splitting a central difference between both sides of the face, in η-direction. It is done as follows:

$$\begin{aligned} \vec{V_\xi} &= \frac{(\phi_{i,j} - \phi_{i-1,j})}{\triangle \xi_{i,j}}, \\ V_\eta &= \frac{1}{2}\left(\frac{\phi_{i,j+1} - \phi_{i,j-1}}{\triangle \eta_{i,j+1} + \triangle \eta_{i,j}}\right) + \frac{1}{2}\left(\frac{\phi_{i-1,j+1} - \phi_{i-1,j-1}}{\triangle \eta_{i-1,j+1} + \triangle \eta_{i-1,j}}\right). \end{aligned} \tag{148}$$

Now when we have the covariant velocity components we need to express the velocity in the Cartesian coordinate system in order to compute the local speed of sound and then the Mach number through this face. The transformation from the covariant coordinate system to the Cartesian coordinate system is done by multiplying the covariant vector by the inverse of the matrix M,

$$[u,\ v]^T = M^{-1}\,[V_\xi,\ V_\eta]^T\,, \tag{149}$$

and the velocity absolute value is,

$$q^2 = \left(u^2 + v^2\right). \tag{150}$$

The local speed of sound is then

$$a^2 = a_\infty^2 + \frac{(\gamma - 1)}{2}\left(V_\infty^2 - u^2 - v^2\right), \tag{151}$$

where $a_\infty = {}^{V_\infty}/_{M_\infty} = 1$. Finally, the Mach number (the ratio of the flow speed to the local speed of sound) M at the half point $(i - {}^1/_2, j)$ is $M = \frac{q}{a}$.

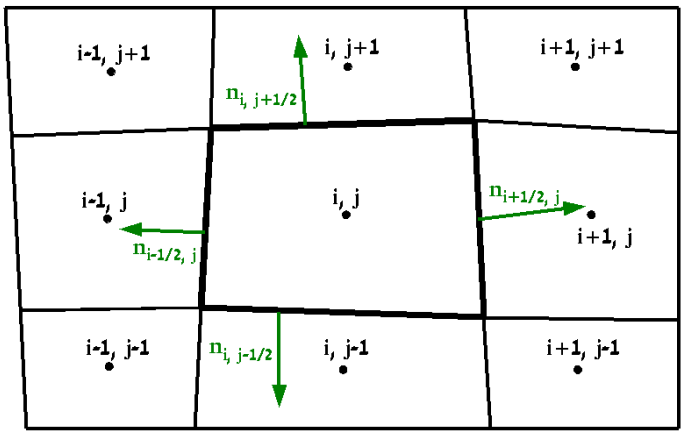


Figure 23: Sub-domain or control volume surrounding a node (i, j).

11.2 Flux calculation — incompressible flow

In this section I will describe in detail how the fluxes of the FPE are discretized. As described in Section 1, the computational space is divided into quadrilateral cells where FPE is solved in each cell separately. Using Cartesian coordinates $(x,\ y,\ z)$ the FPE is written as follows (see Eq. (48)):

$$\frac{d}{dx}(\rho \cdot u) + \frac{d}{dy}(\rho \cdot v) = 0. \tag{152}$$

The velocity components u and v are calculated as the gradient of the potential ϕ.

$$u = \phi_x, \qquad v = \phi_y. \tag{153}$$

The flow is assumed to be uniform in the far field with a Mach number M_∞. At the body surface, "no penetration" boundary condition is applied,

$$\overrightarrow{V} \cdot \overrightarrow{n} = 0, \tag{154}$$

where the product $\overrightarrow{V} \cdot \overrightarrow{n}$ is the velocity component normal to the surface. The density is computed from the isentropic formula (61). The space discretized form of Eq. (152) in the covariant coordinate system (ξ, η) can be written as follows:

$$\frac{d}{d\xi}(F) + \frac{d}{d\eta}(G) = 0, \tag{155}$$

where $F = \rho(\phi_\xi, \phi_\eta)\,\phi_\xi$ and $G = \rho(\phi_\xi, \phi_\eta)\,\phi_\eta$. From the notation above, the fluxes labeled F and G are constructed of terms that contribute to the flux in the ξ and η directions.

The conservatively requirement on Eq. (155) will be satisfied if the scheme can be written under the form:

$$\frac{\left(f_{\left(i+\frac{1}{2},j\right)} - f_{\left(i-\frac{1}{2},j\right)}\right)}{\triangle\xi} + \frac{\left(f_{\left(i,j+\frac{1}{2}\right)} - f_{\left(i,j-\frac{1}{2}\right)}\right)}{\triangle\eta} = 0, \tag{156}$$

where f is the numerical flux at the cell face mid-point $(i-\frac{1}{2}, j)$, $(i+\frac{1}{2}, j)$, $(i, j+\frac{1}{2})$, and $(i, j-\frac{1}{2})$.

In order to simplify the derivation, let us introduce first the incompressible flow equation, which is characterized by a constant density, referenced to a uniform free-stream density ρ_∞, $\rho = \rho_\infty = 1$; this results in

$$\frac{d}{d\xi}(\phi_\xi) + \frac{d}{d\eta}(\phi_\eta) = 0. \tag{157}$$

In a case of a low Mach number flow, an approximation to the entire equation is a discrete nine-point Laplacian, exactly as in the quasi-linear case. Now, let us express ϕ_ξ and ϕ_η as their covariant components,

$$\begin{aligned} \phi_\xi &= F_\xi \hat{\xi} + F_\eta \hat{\eta}, \\ \phi_\eta &= G_\xi \hat{\xi} + G_\eta \hat{\eta}, \end{aligned} \tag{158}$$

where the covariant velocity components, F_ξ, F_η, G_ξ, and G_η, which are calculated at the face $\left(i-\frac{1}{2}, j\right)$, are discretized as follows:

$$
\begin{aligned}
F_\xi &= \frac{3}{4}\frac{(\phi_{i,j}-\phi_{i-1,j})}{\triangle\xi_{i,j}} + \frac{1}{8}\frac{(\phi_{i,j+1}-\phi_{i-1,j+1})}{\triangle\xi_{i,j+1}} + \frac{1}{8}\frac{(\phi_{i,j-1}-\phi_{i-1,j-1})}{\triangle\xi_{i,j-1}},\\
F_\eta &= \frac{1}{2}\frac{(\phi_{i,j+1}-\phi_{i,j-1})}{\triangle\eta_{i,j+1}+\triangle\eta_{i,j}} + \frac{1}{2}\frac{(\phi_{i-1,j+1}-\phi_{i-1,j-1})}{\triangle\eta_{i-1,j+1}+\triangle\eta_{i-1,j}},\\
G_\xi &= \frac{3}{4}\frac{(\phi_{i,j}-\phi_{i,j-1})}{\triangle\eta_{i,j}} + \frac{1}{8}\frac{(\phi_{i+1,j}-\phi_{i+1,j-1})}{\triangle\eta_{i+1,j}} + \frac{1}{8}\frac{(\phi_{i-1,j}-\phi_{i-1,j-1})}{\triangle\eta_{i-1,j}},\\
G_\eta &= \frac{1}{2}\frac{(\phi_{i+1,j}-\phi_{i-1,j})}{\triangle\eta_{i+1,j}+\triangle\eta_{i,j}} + \frac{1}{2}\frac{(\phi_{i+1,j-1}-\phi_{i-1,j-1})}{\triangle\eta_{i+1,j-1}+\triangle\eta_{i+1,j}}.
\end{aligned}
\tag{159}
$$

The parameters $\triangle\xi$ and $\triangle\eta$ are the grid spacing in the ξ and η directions, respectively. The distance separating the cell's centers (i,j) and $(i-1,j)$ is $\triangle\xi_{i,j}$. The horizontal distance separating the cells (i,j) and $(i+1,j)$ is $\triangle\xi_{i+1,j}$. Likewise, the vertical distances that separate (i,j) from $(i-1,j)$ and $(i,j+1)$ are $\triangle\eta_{i,j}$ and $\triangle\eta_{i,j+1}$, respectively. Spacings are computed during the grid generation phase, and are stored as one-dimensional arrays, for later use during the discretization. Since the grid is not uniform we have to take each relevant grid spacing in order to get an accurate discretization. The approximations for the above covariant velocities at the faces $(i+1/2,j)$ and $(i,j+1/2)$ are done in the same way.

In order to get the flux that crosses each face in the cell (i,j), the velocity components must be projected normal to the face through which the flux is computed. Hence, the fluxes are approximated as follows,

$$
\begin{aligned}
F_{i-\frac{1}{2},j} &= \left[\begin{array}{c} F_\xi \\ F_\eta \end{array}\right]_{i-\frac{1}{2},j} \cdot \vec{n}_{i-\frac{1}{2},j},\\
F_{i+\frac{1}{2},j} &= \left[\begin{array}{c} F_\xi \\ F_\eta \end{array}\right]_{i+\frac{1}{2},j} \cdot \vec{n}_{i+\frac{1}{2},j},\\
G_{i,j-\frac{1}{2}} &= \left[\begin{array}{c} G_\xi \\ G_\eta \end{array}\right]_{i,j-\frac{1}{2}} \cdot \vec{n}_{i,j-\frac{1}{2}},\\
G_{i,j+\frac{1}{2}} &= \left[\begin{array}{c} G_\xi \\ G_\eta \end{array}\right]_{i,j+\frac{1}{2}} \cdot \vec{n}_{i,j+\frac{1}{2}}.
\end{aligned}
\tag{160}
$$

Computation of the remaining terms is done in a similar fashion, using the four-step process described above.

11.3 Compressible flow – subsonic flow $(M \leq 1)$

When the Mach number is increased, the density of the fluid changes with respect to the pressure. This case of compressible flow is distinguished from the previous incompressible flow in that the density can no longer be considered constant. Consider the term $\frac{d}{d\xi}\left[\rho\left(\phi_\xi,\phi_\eta\right)\phi_\xi\right]$ in Eq. (155). The terms ϕ_x and ϕ_y in Eq. (61) reflect most of the flow's "dynamic". The discretization approach of these two terms must take into account the value of the Mach number through the face and the flow direction. Consider, for example, the face $(i-1/2,j)$. The term $\frac{d}{d\xi}\left(\rho\left(\phi_\xi,\phi_\eta\right)\phi_\xi\right)$ in Eq. (155) may be computed at the half node $(i-1/2,j)$ as follows:

1. First, the discretization of ϕ_ξ (fluxes of the Laplacian) that were presented in the previous subsection, holds.

2. Next, the Cartesian velocity components u and v, the contravariant velocity components V_ξ and V_η, the local speed of sound a, and the local Mach number M, are computed at the half node $(i - 1/2, j)$.

3. Now, when the Local Mach number is available across the face $(i - 1/2, j)$ and the flow direction is known, we can decide how to discretize the velocity ϕ_ξ (recall that we are dealing with the flux through the face $\left(i - \frac{1}{2}, j\right)$ in the equation for the density

$$\rho(\phi) = \rho_\infty \left[\frac{(\gamma - 1)}{2} \left(V_\infty^2 - \phi_\xi^2 - \phi_\eta^2 \right) \right]^{\frac{1}{(\gamma-1)}}, \tag{161}$$

where the velocity vector of the flow through this face has two covariant components as follows:

$$V_{cov} = V_\xi \hat{\xi} + V_\eta \hat{\eta}. \tag{162}$$

The discretization of ϕ_ξ depends on the direction of the flow relative to the vector $\hat{n}$, normal to the face, as sketched in Figure 24.

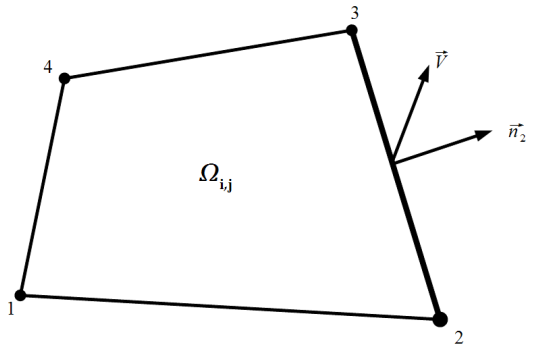


Figure 24: Definition of the parameter $\delta = \hat{V} \cdot \hat{n}$

The parameter delta is defined as,

$$\delta = \hat{V} \cdot \hat{n}, \tag{163}$$

where $\hat{V}$ and $\hat{n}$ are unit vectors of the flow's velocity and the face's normal, respectively. The way we choose to discretize the velocity terms ϕ_x and ϕ_y is determined by the parameter δ. For example, when $\delta = 1$ the flow direction is normal to the face 2-3, and it results in a "wide" approximation in the ξ-direction and "narrow"

approximation in the η-direction. The case of $\delta < 0$ indicates that the flow direction is opposite to the way that the grid's indices are defined. It is a typical situation when solving the flow field through a cylinder, or airfoil. Further details and examples are presented in Section 13.

The covariant velocity components through face $2-3$, V_ξ, and, V_η, are derived by central differencing for a flow in a general direction as follows:

$$\begin{aligned}
\vec{V}_\xi &= \left(1 - \frac{1}{2}\delta^2\right)\frac{(\phi_{i,j} - \phi_{i-1,j})}{\triangle\xi_{i,j}} + \frac{1}{4}\delta^2\frac{(\phi_{i,j+1} - \phi_{i-1,j+1})}{\triangle\xi_{i,j+1}} + \frac{1}{4}\delta^2\frac{(\phi_{i,j-1} - \phi_{i-1,j-1})}{\triangle\xi_{i,j-1}}, \\
\vec{V}_\eta &= \frac{1}{4}\left(\frac{\phi_{i,j+1} - \phi_{i,j}}{\triangle\eta_{i,j+1}}\right) + \frac{1}{4}\left(\frac{\phi_{i-1,j} - \phi_{i-1,j-1}}{\triangle\eta_{i-1,j}}\right) \qquad if \qquad \hat{V}_y \geq 0, \\
\vec{V}_\eta &= \frac{1}{4}\left(\frac{\phi_{i,-1j+1} - \phi_{i-1,j}}{\triangle\eta_{i-1,j+1}}\right) + \frac{1}{4}\left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\triangle\eta_{i,j}}\right) \qquad if \qquad \hat{V}_y < 0,
\end{aligned} \tag{164}$$

where the velocity vector in the Cartesian coordinate system is defined as $\overrightarrow{V} = V_x\hat{x} + V_y\hat{y}$. The terms $\hat{x}$ and $\hat{y}$ are unit vectors in the x and y directions, respectively.

4. When the density ρ and the covariant velocity ϕ_ξ are available at the half node $(i - 1/2, j)$, the flux can be computed by

$$f_{i-1/2.j} = \rho\left(\phi_\xi, \phi_\eta\right)\phi_\xi. \tag{165}$$

5. Computation of the remaining fluxes at $(i + 1/2, j)$, $(i, j + 1/2)$, and $(i, j - 1/2)$ is done in a similar fashion, using the four-step process described above.

11.4 Compressible flow – supersonic flow $(M > 1)$

In the supersonic region the equation changes type from elliptic to hyperbolic, and therefore, a "wide" upwind difference should be used to approximate the derivatives in the streamwise direction. In the construction of the discrete approximation to the conservation form, it is necessary to switch to upwind differencing. Consider again the flux through the face $(i - 1/2, j)$. The flux in the ξ-direction is approximated as follows:

1. A central discretization of the Laplacian holds also in this case.

2. Next, according to relations presented in Section 12.1, the Cartesian components of velocity u and v, the contravariant component of velocity $V_{\xi n}$ and $V_{\eta n}$, the local speed of sound a, and the Mach number M, are computed at the half node $(i - 1/2, j)$.

3. The discretization of the velocities ϕ_x and ϕ_y in Eq. (61) is done in the streamwise direction while taking into account an upwind approximation as the flow exceeds $M = 1$. The final formula for ϕ_x and ϕ_y is written as follows:

$$\begin{aligned}
\phi_x^2 &= \frac{1}{M_{i,j}^2}V_\xi \cdot V_{\xi n} + \left(1 - \frac{1}{M_{i,j}^2}\right)V_\xi^u \cdot V_{\xi n}^u, \\
\phi_y^2 &= \frac{1}{M_{i,j}^2}V_\eta \cdot V_{\eta n} + \left(1 - \frac{1}{M_{i,j}^2}\right)V_\eta^u \cdot V_{\eta n}^u.
\end{aligned} \tag{166}$$

The superscript u denotes an upwind approximation. Since the equation changes type from elliptic to hyperbolic, therefore, a "wide" upwind difference should be used. As an illustration, when the flow is grid aligned, the fluxes at the half node $(i - 1/2, j)$ are discretized, in a stencil notation, as follows:

$$V_\xi = \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} \\ -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix}_{i,j}, \qquad V_\eta = \begin{bmatrix} & \\ -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}. \tag{167}$$

As one can see, the upwind differencing is introduced smoothly since $\frac{1}{M^2} \to 1$ as $M \to 1$. If the upwind differencing were introduced directly as $M_\infty > 1$, there would be a sudden change in the representation of the Laplacian term $\nabla^2\phi$, which does not vanish when $M_\infty = 1$. A scheme of this type was tested and found to be much less efficient.

Any numerical scheme used to solve the FPE for the potential ϕ must satisfy the domain of dependence. Since the characteristics are symmetric about the velocity vector, both the ξ and η derivative terms must be properly shifted when solving the FPE. The discretization of V_ξ and V_η is already was presented in the previous subsection. However, the upwind approximations V_ξ^u and V_η^u in a general direction is done as follows: In case of $\delta \geq 0$,

$$\begin{aligned}
V_\xi^u &= \left(\frac{1}{2}\delta^2\right)\frac{\phi_{i-1,j} - \phi_{i-2,j}}{\triangle\xi_{i-1,j}} + \left(\frac{1}{4}\delta^2\right)\frac{\phi_{i-1,j-1} - \phi_{i-2,j-1}}{\triangle\xi_{i-1,j-1}}, \\
&+ \left(\frac{1}{4}\delta^2\right)\frac{\phi_{i-1,j+1} - \phi_{i-2,j+1}}{\triangle\xi_{i-1,j+1}} + \frac{1}{2}\left(1 - \delta^2\right)\frac{\phi_{i,j-1} - \phi_{i-1,j-1}}{\triangle\xi_{i,j-1}}, \\
&+ \frac{1}{2}\left(1 - \delta^2\right)\frac{\phi_{i,j} - \phi_{i-1,j}}{\triangle\xi_{i,j}}, \\
V_\eta^u &= \frac{1}{2}\frac{\phi_{i-1,j} - \phi_{i-1,j-1}}{\triangle\eta_{i-1,j}} \qquad if \qquad \hat{V}_y \geq 0, \\
V_\eta^u &= \frac{1}{2}\frac{\phi_{i-1,j+1} - \phi_{i-1,j}}{\triangle\eta_{i-1,j+1}} \qquad if \qquad \hat{V}_y < 0.
\end{aligned} \tag{168}$$

If $\delta < 0$, then

$$\begin{aligned}
V_\xi^u &= \left(\frac{1}{2}\delta^2\right)\frac{\phi_{i+1,j} - \phi_{i,j}}{\triangle\xi_{i+1,j}} + \left(\frac{1}{4}\delta^2\right)\frac{\phi_{i+1,j-1} - \phi_{i,j-1}}{\triangle\xi_{i+1,j-1}}, \\
&+ \left(\frac{1}{4}\delta^2\right)\frac{\phi_{i+1,j+1} - \phi_{i,j+1}}{\triangle\xi_{i+1,j+1}} + \frac{1}{2}\left(1 - \delta^2\right)\frac{\phi_{i,j+1} - \phi_{i-1,j+1}}{\triangle\xi_{i,j+1}}, \\
&+ \frac{1}{2}\left(1 - \delta^2\right)\frac{\phi_{i,j} - \phi_{i-1,j}}{\triangle\xi_{i,j}}, \\
V_\eta^u &= \frac{1}{2}\frac{\phi_{i,j+1} - \phi_{i,j}}{\triangle\eta_{i,j+1}} \qquad if \qquad \hat{V}_y \geq 0, \\
V_\eta^u &= \frac{1}{2}\frac{\phi_{i,j+1} - \phi_{i,j}}{\triangle\eta_{i,j+1}} \qquad if \qquad \hat{V}_y < 0.
\end{aligned} \tag{169}$$

In these approximations the velocity is assumed to be from left to right, namely $\hat{V}_x > 0$.

4. When ϕ_x^2 and ϕ_y^2 are calculated, the density ρ is available at the half node $(i - 1/2, j)$. Then, the flux (in ξ-direction) can be calculated by Eq. (165).

5. Computation of the remaining fluxes at $(i + 1/2, j)$, $(i, j + 1/2)$, and $(i, j - 1/2)$ is done in a similar fashion, using the four-step process described above.

11.5 Boundary conditions

A distinction of fluid flows in two different cases depends solely on the boundary condition. The numerical treatment of the boundary conditions requires particular care. An improper implementation can result in an inaccurate simulation of the real system. In addition, the stability and convergence rate of the scheme can be adversely influenced. For completeness we will introduce first the two common types of boundary conditions, Dirichlet and Neumann.

11.5.1 Dirichlet condition

The potential is prescribed along the boundaries. Practically, the values of the ghost cells are fixed. Only the interior cells are relaxed, so the boundary conditions are satisfied on all the levels.

11.5.2 Neumann condition

The normal derivative of the potential ϕ (namely the velocity components u and v) is specified along the boundary. Note that the relaxation (which is performed on interior cells only) should change the boundary value, not so that the corresponding boundary condition is satisfied exactly, but to smooth the error along the boundary. Each relaxation sweep is followed by an update of the ghost cells in order to satisfy the boundary condition.

Further types of boundary conditions that were applied in several case studies presented in this research are as follows:

- Inflow and outflow internal flows,
- Solid wall,
- Coordinate cut,
- Periodic boundary.

The numerical treatment of the first two cases will be covered in the following lines since it takes place in all the flow cases that were tested in this research work. The next two cases will be covered in the relevant application. For further information of boundary conditions the reader is referred to Blazek [90, 12]. At the boundary, the information will reflect either the free-stream conditions, imposed conditions, or conditions inside the computational domain. The boundary conditions need to be applied so that the unknown variables at the boundary reflect the correct information. Since the 2D FPE is being solved here, only the velocity components will need to be specified at each boundary. For this purpose we use the dummy cells.

11.5.3 Dummy cells

In general, boundary conditions can be applied to either boundary faces or exterior boundary cells, which lie outside the boundary layer. These exterior cells are often called dummy cells or ghost cells. The dummy cells are additional layers of grid cells outside the physical domain. This is sketched in Figure 25 for the case of a $2D$ structured grid. The whole computational domain is surrounded by two layers of dummy cells (dashed lines). The cells are virtual (ghost cells), although geometrical quantities such as areas and flux vectors are associated with them. The purpose of the dummy cells is to simplify the computation of the fluxes along the boundaries. This can be achieved by the possibility of extending the stencil of the discretization scheme beyond the physical boundaries. The number of dummy cells must be such that the part of the stencil outside the physical domain is completely covered. The conservative variables values in the dummy cells are obtained from the boundary conditions.

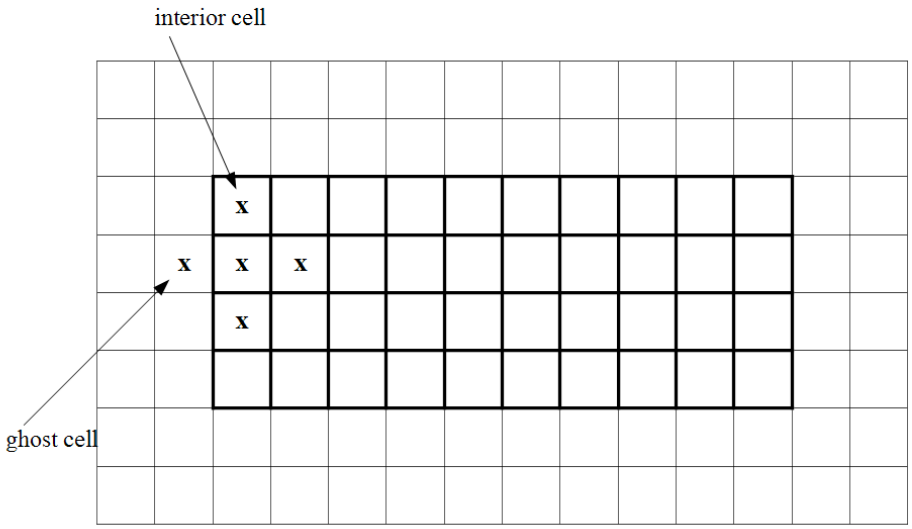


Figure 25: Two layers of dummy cells around the 2D computational domain.

11.5.4 Inflow / outflow

The first type of boundary condition to be discussed is a flow entering or leaving the domain, commonly referred to as inflow and outflow conditions. Since the 2D FPE is being solved here, the velocity components u and v are specified as either the free-stream condition or a given velocity profile. The treatment of the outflow boundary condition is the same.

11.5.5 Solid wall

The second type of boundary is wall or solid surface. Figure 26 presents the solid wall boundary condition for the cell-centered scheme. Since in this work we are dealing with an inviscid flow, the fluid slips over the surface. This is known as the nonpenetrating condition. There is no friction force, so the velocity vector must be tangent to the surface,

$$\frac{\partial \phi}{\partial n} = 0 \tag{170}$$

at the surface, where $\overrightarrow{n}$ denotes the unit normal vector at the surface. Hence the contravariant velocity $\overrightarrow{V}$ is zero at the wall. When the grid is orthogonal, the implementation of this boundary condition is relatively easy. But since the curvilinear coordinate system is wrapped around the body, which results in a nonorthogonal grid, this condition is more difficult to implement. In order to attain this value we have to answer the following question: what is the value of the dummy cell that we have to impose in order to get a zero contravariant velocity, $\vec{V}_{\xi n} = 0$?

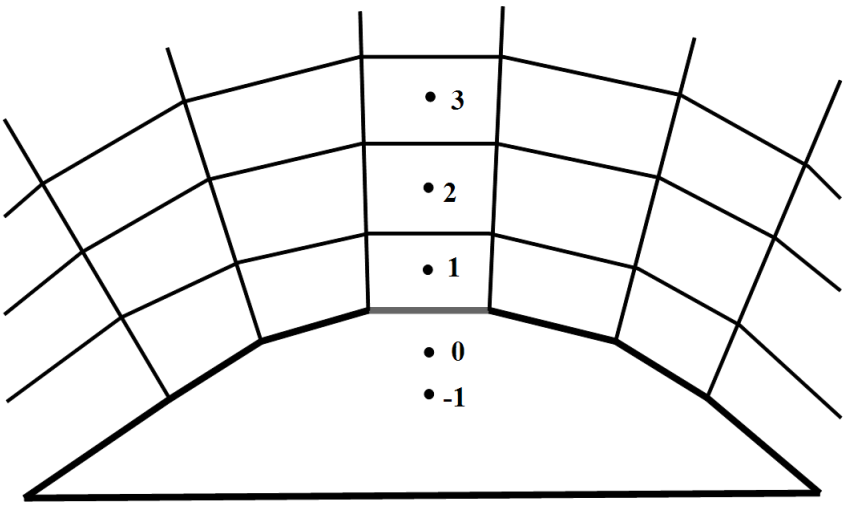


Figure 26: Solid wall boundary condition for the cell-centered scheme. Dummy cells are denoted as 0 and -1.

Since the discretization of the velocities is done in a covariant coordinate system, we can build an equation for the contravariant velocity, $\vec{V}_{\xi n}$. By combining Eqs. (142) and (143) we get a system that transforms the velocity vector from the covariant to the contravariant coordinate system,

$$\begin{bmatrix} V_{\xi n} \\ V_{\eta n} \end{bmatrix} = \left(M^T\right)^{-1} M^{-1} \begin{bmatrix} V_{\xi} \\ V_{\eta} \end{bmatrix}. \tag{171}$$

According to the following example, the equation for the contravariant velocity has to be solved only in η-direction. So substituting the relation for the covariant velocity we get the following relation:

$$\begin{bmatrix} V_{\xi n} \\ V_{\eta n} \end{bmatrix} = \left(M^T\right)^{-1} M^{-1} \begin{bmatrix} V_\xi \\ \frac{\phi_{i,j}-\phi_{i,j-1}}{\triangle \eta_{i,j}} \end{bmatrix}. \tag{172}$$

As one can see, the equation for $\vec{V}_{\eta n}$ (the second equation) is an algebraic equation with one unknown – $\phi_{i,j-1}$. Expanding the equation for $\vec{V}_{\eta n}$ we get

$$V_{\eta n} = C_1 V_\xi + C_2 \left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\triangle \eta_{i,j}}\right) \to 0, \tag{173}$$

where C_1, C_2, and $\vec{V}_\xi$ are computed as follows:

$$C_1 = \frac{-\left(\hat{\eta}_y \hat{\xi}_y + \hat{\eta}_x \hat{\xi}_x\right)}{\left(\hat{\xi}_x \hat{\eta}_y - \eta_x \xi_y\right)^2} \qquad C_2 = \frac{-\left(\xi_y^2 + \xi_x^2\right)}{\left(\hat{\xi}_x \hat{\eta}_y - \eta_x \xi_y\right)^2}, \tag{174}$$

$$V_\xi = \frac{1}{2}\frac{(\phi_{i+1,j} - \phi_{i-1,j})}{(\triangle \xi_{i+1,j} + \triangle \xi_{i,j})} + \frac{1}{2}\frac{(\phi_{i+1,j-1} - \phi_{i-1,j-1})}{(\triangle \xi_{i+1,j-1} + \triangle \xi_{i,j-1})}. \tag{175}$$

In this way all the dummy cells along the solid wall boundaries were evaluated.

12 Applications

The efficiency of the AMG method in solving the second order nonlinear FPE in the conservation form for various compressible flow problems is studied in this section. All the results were obtained with the algorithms described in the previous sections. Several observations are in order here, related to the AMG setup phase and solution phase in the various flow problems:

1. Generally, strong connectivity is defined by $\varepsilon = 0.25$. When the problem is subsonic, and the velocity is quite far from the sonic case, we find that there is no advantage in using the dynamic threshold in the coarsening process (described in Section 4). Instead, the fixed threshold does the work quite well, in terms of convergence properties and complexities. But in some cases, especially when the matrix A^m is not an M-matrix, the influence of the fixed threshold as a measure of strength between the matrix variables becomes dominant. It causes not only the smoothing to be less effective but also interpolation to be less accurate (since fewer points are involved in the interpolation process). It results in a much slower convergence rate (above 0.5), in the cases where we have convergence at all. The convergence can be improved, for instance, by applying more smoothing steps in each V-cycle. However, this would substantially increase the cycle cost and more importantly, won't improve the interpolation process. In such cases the dynamic threshold is applied. However, the dynamic threshold is implemented only when no other alternative is available. Practical experience has clearly shown that the dynamic threshold tends to increase the number of points on the coarse-levels. It results in a slightly increased operator complexity and

especially grid complexity. The alternative, namely to try to improve the increased complexity by modifying the truncation of the coarse-level operator (see remark 5 in this Section), will be generally much more expensive. Nevertheless, in light of the strong nonlinear cases involved, and the discontinuities in the matrix's coefficients, the cost that we pay for the convergence properties is quite acceptable. In fact, the dynamic threshold together with the direct approximation approach (discussed in Section 8 for construction of the coarse-level and restriction operator) is a "brute force" approach to improve the AMG performance.

2. The coarse-level and restriction operators are construed based on the strong and weak connections between the matrix A^* entries. The coarse-point selection algorithm proceeds in two passes. We first make an initial division of the grid points by choosing a preliminary partition into C- and F-points. Once the initial assignments have been made, a second pass process is applied in order to satisfy the coarsening requirements as described in Section 8. In this way the coarse-level and the restriction operator are constructed. Next the interpolation operator is constructed based on the algorithm described in Section 8. A question to be asked is how many coarse-levels are needed in order to solve the problem? Normally it is unknown how many levels of coarsening to execute from the beginning. The base case of the algorithm is satisfied when the current level is sufficiently coarse. This is a broad term and basically there are several choices. For instance, one could check for the number of data points or simply limit the number of levels. In this algorithm the first approach is chosen. The number of points at the coarsest level is limited to about $\sim 1\%$ of the total number of points on the finest level. The coarsening process, which includes the construction of the coarse-levels and the restriction operators, continues until this criterion is satisfied. In each coarse-level, the interpolation operator is constructed.

3. Our focus, in the problems involved, is to reduce the L_2-norm of the residual below 10^{-10}, with a convergence rate of less than an order of magnitude per V-cycle. However, judging just the convergence behavior is not sufficient for having a robust AMG. Definitely, we had to take into account the computational work and memory requirements. The computational work is determined by the operator complexity C_L, and the convergence rate C_f. Only if both quantities are bounded as the problem is increased do we have an optimal performance. However, compared to problems on very regular meshes, a decrease in the AMG convergence is expected in the case of irregular meshes. The main reason is the interpolation, which tends to be close to geometrical interpolation in regular meshes. The influence of the irregularity of the grid and its dimension on the convergence behavior is problem dependent. In such cases there is a small increase in the convergence factor as well as in the operator and grid complexities. This situation is bounded and can be controlled by the dynamic threshold parameter. By experience, the effects mentioned are very limited for the problems that we have tested in this work.

4. If the flow speed in a given problem reaches a local Mach number, let us

say below 0.7, the coefficients in the matrix A^* are nearly "frozen" during the solution process. In this situation, the setup phase, which includes the construction of the matrices and transfer operators (restriction and interpolation), was performed only once, at the beginning of the algorithm, and no extra updates of the matrices A, $\tilde{A}$, and A^* were needed. This is the situation in most of the problems that were tested. When a transonic flow is involved, it results in a strong anisotropy (near the boundary) and the treatment of such a problem is more difficult and requires special attention. The nonlinearity is more dominant and 2-3 successive V-cycles are followed by a new setup phase in order to achieve the desired convergence rate (the frequency in which a setup-phase is performed is problem-dependent). This is done since the non-linearity is solved on the fine-level only. Since the matrices considered are constructed of nonlinear coefficients, they must be updated. Performing the setup-phase is expensive in terms of storage, computation, and time, bur it is very essential in order to achieve convergence. The setup phase can be performed after 1-2 V-cycles and it is problem-dependent. This process is repeated until the residual reaches a sufficient level. A possible way to improve this process is to update only the points belonging to the region of the strongest anisotropy. However, this approach was not actually implemented.

5. Through all the problems presented in this research work, a direct approximation based on the fine-grid operator A^m to construct the coarse-level matrix A^{m+1} and the restriction operator I_m^{m+1} was applied (see details in Section 8). In practice, this process, which includes several operations on the matrix A^m entries, results in a significantly less sparse coarse-level matrix. Most of the entries, however, are small relative to the diagonal, and therefore, can be ignored without seriously sacrificing convergence. Consequently, in order to keep the resulting coarse-level operator as sparse as possible, the construction of the coarse-level operator should always be combined with a reasonable truncation. This is performed immediately when the coarse-level operator is computed, by eliminating the small values relative to the diagonal according to the following: $\frac{|a_{i,j}|}{|a_{i,i}|} \leq \mu$, where $\mu = 10^{-4}$. Practically, all those entries of the coarse-level matrix that satisfy this condition are added to the diagonal, in order to preserve the row sum. This process is repeated for all the coarse-level operators and it worked perfectly well for all the problems that we have tested.

   It is very important to mention here that, since the coarse-level operator is constructed by direct approximation, the truncation of the small entries can be implemented directly on the coarse-level only (and not as an interpolation truncation as can be done in the Galerkin operator). This process is dangerous since in some circumstances, an "important" entry can be eliminated and if not applied with great care, may cause divergence in practice. As a comparison, if the Galerkin operator was applied (at least for the subsonic case where the equation is elliptic and we can still reach convergence) the truncation of the small entries would be implemented to the interpolation operator and not to the Galerkin operator directly. The reason is that in the worse case the convergence may slow down but no divergence can oc-

cur. However, Galerkin operator is not relevant for the following "tough" aerodynamic problems that we are facing.

6. It should be clear that this AMG algorithm is designed to deal with problems characterized with various flow conditions. This directly refers to the type of a given matrix A. However, we are not really interested in, for instance, an algorithm version for one particular A only (for instance an M-matrix) but rather in having a uniform AMG setup if the matrix A ranges over some reasonable class of matrices. A reasonable class may consist of the discretization matrices of both elliptic and hyperbolic equations discretized on an irregular grids.

7. The AMG components which discussed in Section 8, generally are not designed to be optimal. Unfortunately, they will always be constructed on the basis of compromises between numerical work and overall efficiency. In particular cases, when the flow is subsonic (which is characterized by an elliptic operator) with a regular or an irregular grid, we show that the convergence is independent of the problem size and is as fast and expensive as we wish. Actually, the convergence rate is not generally the problem; in fact, the AMG can always be forced to converge rapidly, but rather there is the trade-off between convergence, numerical work, and memory requirements. In this respect, while constructing the AMG algorithm, low memory approaches were of particular interest, even if the reduced memory requirement causes a degradation of the convergence properties. In any case, since the problems that we have faced with are characterized by an extreme anisotropy, with irregular grids, shock waves, and strong nonlinearity, we focused on the following requirements: The operator complexity must not be significantly larger than 3.0 and the convergence factor around an order of magnitude. We will see that these requirements are satisfied in most of the test cases considered.

8. This algorithm is far from being optimal. It is not being designed for the greatest efficiency but rather to be as uniform as possible, that could cover most of the relevant cases regarding the FPE. The AMG components are constructed on the basis of compromises between numerical work and overall efficiency. Our main interest was to use the new algorithmic components that have already been discussed and implement them as fixed strategies. They are not adjusted to the particular requirements of a given problem. For example, if the coarse-level and restriction operator are constructed according to Chang [93] and not by the Galerkin condition, it is always applied in all the flow conditions considered in the relevant problem, although the Galerkin condition can produce much better results (in a very specific flow conditions, for instance, subsonic flow). Similarly, if a dynamic threshold is applied, it is done in all the coarse-levels. There are situations where a local application may give improved convergence properties with much lower cost and memory. For example, by applying a fixed threshold only to the fine-level and for the rest of the coarse-levels a dynamic threshold is applied. Similarly, aggressive coarsening, which can improve the memory requirements on the expense of slower convergence rate. Thus, there is much

room for quite substantial optimizations and one may think of various modifications and improvements relating to the AMG's coarsening strategy and smoothing. However, by our experience, the algorithm in its current form is very flexible, robust, and efficient in practice.

9. For analyzing the AMG performance, the timing for constructing the setup phase and the solving phase was not taken into account as a parameter for quality of the code. Since the timing depends on both the machine and the compiler, it has to be judged with care. Comparisons typically change from machine to machine.

In order to make the description of the following applications clearer, and more self-contained, we repeat the most relevant aspects described above. References to previous sections are kept to a minimum.

12.1 Overview of the solution procedure

The basic steps involved in the iterative solution of the FPE while applying the AMG method in a curvilinear grid will now be covered.

1. The first step is to construct the structured curve linear grid. The grid must be smooth, with adequate resolution near the body surfaces.

2. Compute and store the following geometrical measurements of each cell in the computational domain:

   (a) Face vector S.

   (b) Face length $\triangle S$.

   (c) Unit normal vector at each face $\hat{n}_m$.

   (d) Transformation matrices M and M'.

3. For simplicity, the initial conditions were assumed to be uniform flow in the free-stream direction, at the free-stream velocity V_∞. First, the velocity gradients are derived in a covariant coordinate system. Second, we know that the free-stream velocity is in the x-direction. Since the derivatives are computed in the covariant coordinate system (ξ, η), we have to find the covariant velocities. The equations in which the covariant velocities are computed in each cell are written as,

$$\begin{bmatrix} V_\xi \\ V_\eta \end{bmatrix} = M \begin{bmatrix} V_\infty \\ 0 \end{bmatrix}. \tag{176}$$

   Replacing the velocity by its derivative

$$V_\xi = \frac{\phi_{i,j} - \phi_{i-1,j}}{\triangle \xi_{i,j}}, \tag{177}$$

   the expression for the potential at the cell's center is

$$\phi_{i-1,j} = \phi_{i,j} - \triangle \xi_{i,j} M_{1,1} V_\infty, \tag{178}$$

where $M_{1,1}$ is the first coefficient of the matrix M (transfers a velocity vector from the Cartesian to the covariant coordinate system) which is already computed and stored. Therefore, the potential is known at any point in the field. This process of applying an initial condition works fine even for the transonic flow cases where the nonlinearity is significant.

4. The setup phase – the matrices A, and, $\widetilde{A}$ are constructed by a linearization process, then we get $A^* = A \cdot \tilde{A}$. All operations performed by the AMG process, in a given AMG cycle, are based upon the matrix A^*.

5. The coarse-level and restriction operators are constructed based on the strong and weak connections between the matrix A^*entries. The coarse point selection algorithm proceeds in two passes for construction of the coarse-level and restriction operator. Next the interpolation operator is constructed based on the algorithm described in Section 8.

6. The solving phase – a $V(2,2)$ cycle with SGS smoothing steps is applied recursively on the coarse-levels. Only on the finest level is a direct relaxation is applied (without using the matrix formation) in order to have the ability to apply local smoothing sweeps where it is necessary. In the coarse-levels the SGS is applied only in the matrix form. The coarsest level was solved iteratively exactly as the other coarse-levels. Each direct relaxation on the fine-level is followed by an update of the ghost points in order to ensure that the boundary conditions are satisfied.

7. In order to understand the simulation results and verify that the physical problem is solved, under the limitations of the potential flow model several flow characteristics such as the velocity field, Mach number, and pressure coefficient were computed in each cell. Since the potential value is defined at the cell's center and the presentation of the flow characteristics is done on the vertices of the grid, an interpolation is performed. The velocity at a given vertex is an average of the flow velocities through the faces that surround it. As the velocity is known, the local speed of sound and the local Mach number can be calculated for each vertex.

12.2 Numerical experiments

Several two-dimensional flow calculations have been performed to test the performance of the AMG method implemented within the FPE within the body-fitted structured grid finite volume context. The test-cases were chosen to address two major requirements: First, the flow model problem has to agree with the potential flow limitations. Second, it should allow to examine capability of the code to deal with irregular structured grids together with an equation that becomes extremely anisotropic near the sonic case and changes type to hyperbolic in the supersonic flow regime. The following model problems certainly do not give the full picture, but they most easily allow investigation of the AMG's asymptotic behavior as well the ability to deal with various specific aspects such as extreme anisotropy (which can be obtained by the grid or the equation itself), nonlinearity, discontinuities, and shock waves. Two-dimensional solutions will be given for the following problems:

1. Channel with a circular bump.
2. A circular cylinder with and without circulation.
3. NACA-0012 airfoil. The simulation includes two angles of attack, $\alpha = 0°$and $\alpha = 1.25°$.
4. NACA-2822 airfoil with $\alpha = 0°$.
5. Convergent-divergent diffuser.
6. Nozzle.
7. Solid propellant rocket engine.

The problems were tested in subsonic and transonic flow regimes in different grid resolutions. All the computational data in the subsonic flow regime is two orders accurate in space and first order accurate in the supersonic region. All the experiments have been run on a computer with a 2.4 GHz Pentium 4 processor and 4 GB of RAM. For each experiment, we report the following data.

We consider several measures of the effectiveness of the algorithm. Our focus is on solving the FPE in each case to a high degree of accuracy, namely by reducing the residual by at least ten orders of magnitude. The rate by which the residual reached a sufficient level is clearly reflected by the convergence factor. As the convergence rate is reduced so is the number of iterations that are required in order to decrease the residual to a sufficient level. The grid and the operator complexities were also computed as a measure of the algorithm's efficiency.

The column denoted by $\|R^m\|_2$ reports the residual L_2-norm after each V-cycle. For each V-cycle the columns denoted by C_f report the convergence factor, which is the ratio of $\|R^m\|_2$ between successive V-cycles. In the row denoted by C_Ω, the grid complexity is reported, which is the total number of grid points on all the levels, divided by the number of grid points on the finest level. The operator complexity is reported as C_L, associated with the V-cycle for every run (defining operator complexity in the usual sense (see Section 5), as the ratio of the number of entries stored in all matrices on all levels divided by the number of entries stored in the finest-level).

In order to define the AMG's setup, various parameters have to be defined. The following default settings were used throughout the calculations, unless explicitly stated otherwise:

- The coarsening process including construction of the restriction operator is done according to Algorithm 2, described in Section 8 for all the problems that follow. Although Algorithm 1 is simpler to implement, there is a significant advantage to the second method over the first, in the supersonic flow regime, where the discrete operator does not lead to an M-matrix.
- The second pass process was applied only for the fine-level, in order to satisfy the interpolation requirements.
- Strong connectivity is defined by the fixed threshold $\varepsilon = 0.25$. The dynamic threshold is used where the fixed threshold fails and it is problem dependent.

- By default we use a symmetric Gauss–Seidel, two pre- and two post-smoothing steps being the default.
- The type of interpolation used is according to that described in Section 8.
- Coarsening is terminated as the number of points at the coarsest level drops below $\sim 1\%$ of the total number of points on the finest level. In the case where a supersonic speed is involved, the coarsening is terminated according to the details described in Section 8.

12.3 Channel with a bump

This is the first simple representative example of an internal flow while solving the FPE in the conservation form. It consists of a channel of height L and length $3L$. Along the bottom wall there is a circular arc of length L and thickness $0.1L$. An H-grid is constructed, based on the algorithm of Blazek [90], using 96 cells in the x-direction and 32 cells in the y-direction. Also coarser grids of 24×8 and 48×16 were computed based on this algorithm. The mesh used for the channel flow test case is presented in Figure 27.

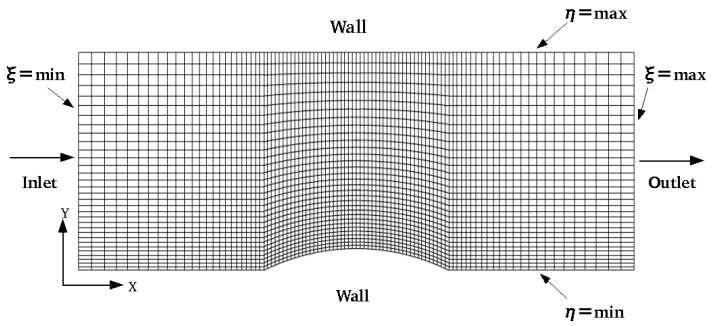
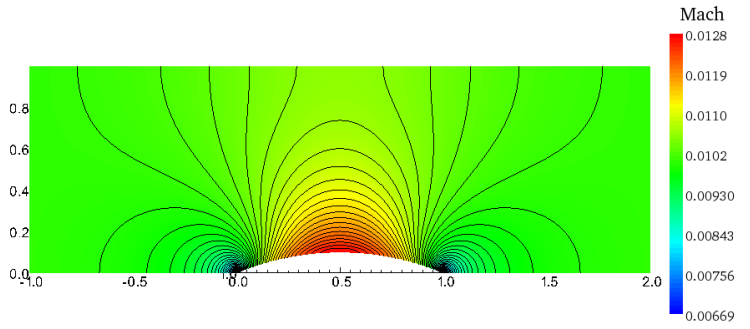
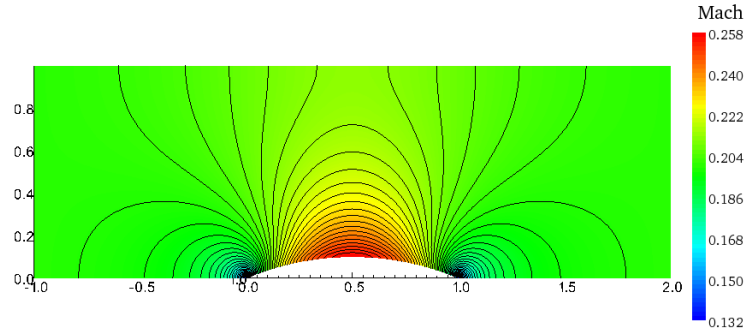


Figure 27: Mesh 96×32 used for the channel flow test case.

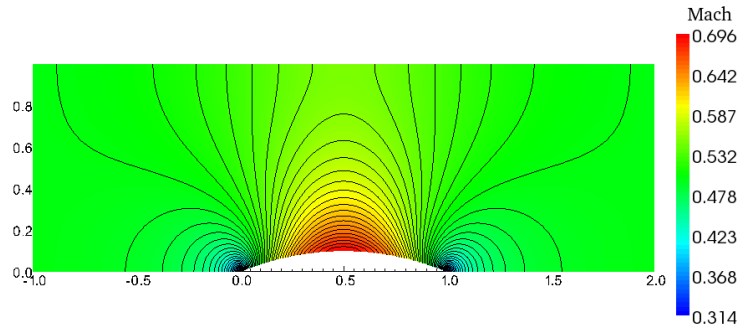
A constant velocity and Mach number as inlet and outlet flow conditions were imposed. At the top and bottom, a solid wall boundary condition was applied, as detailed in Section 12. A uniform flow velocity is applied as an initial condition. The Mach number isolines and color maps are shown in Figure 28. The problems were solved with the following Mach numbers: 0.01, 0.25, 0.5, and 0.64. One isoline has been drawn on the color map every 0.013 ranging from 0.0 to 0.65. It is clear that within the low Mach number range the solution does not greatly depend on the Mach number and the Mach isolines are practically identical. When increasing the M_∞, the compressibility effects become more dominant and for an incident velocity of $M_\infty = 0.64$ a supersonic region is terminated by a shock (see Figure 28(d) appears above the bump. The pressure coefficient for the four cases described above is presented in Figure 29. The shock jump is clearly visible in the pressure distribution when $M_\infty = 0.64$. The results compare very well with Mach contour distributions in the literature [94, 12, 17].



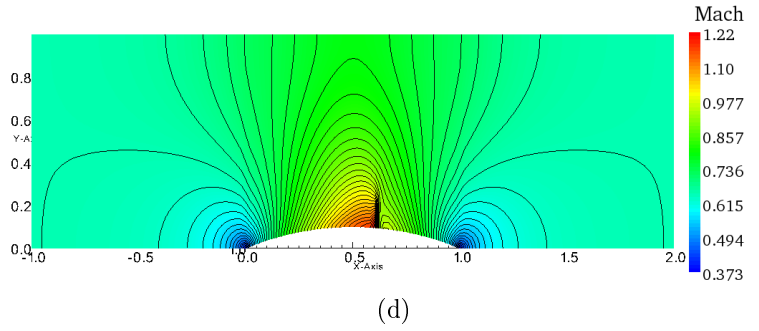
(a)



(b)



(c)



(d)

Figure 28: Mach number isolines computed on the (96×32) mesh for the following incident Mach numbers: a) $M_\infty = 0.01$, b) $M_\infty = 0.25$, c) $M_\infty = 0.5$, d) $M_\infty = 0.64$. Observe the shock appearing at $M_\infty = 0.64$.

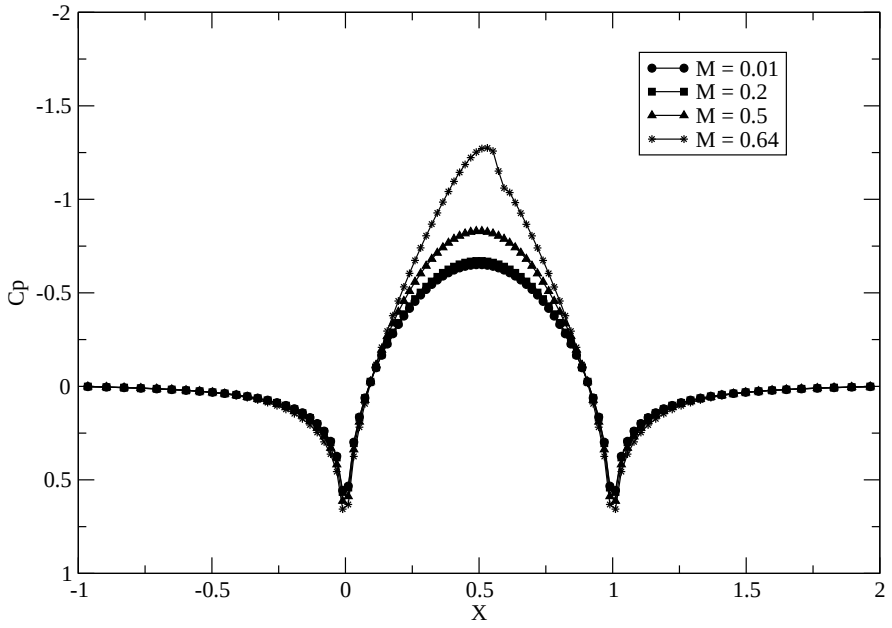
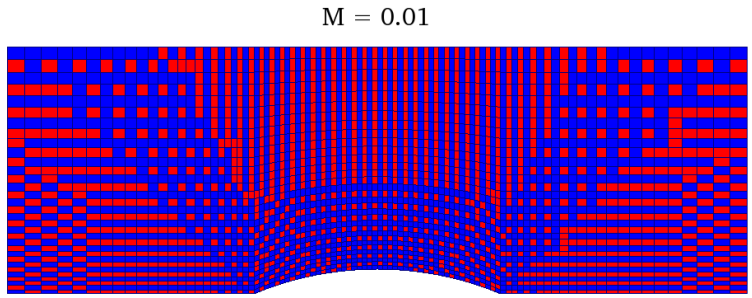


Figure 29: The pressure coefficient C_p calculated at the bottom wall for $M_\infty = 0.01 - 0.64$. Observe the shock appearing at $M_\infty = 0.64$.

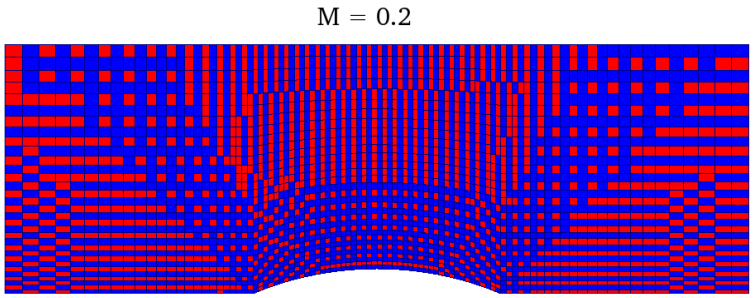
AMG performance

The first coarse-level for each case described above is sketched in Figure 30. In fact, these coarsening patterns are very interesting, and several observations are in order here. First, the case of $M_\infty = 0.01$ is characterized by a practically incompressible flow that results in nearly an isotropic equation. So, we would expect that the coarse points would be distributed uniformly at the entire flow field. But as can be seen in Figure 30(a) this is not the case. The reason for that is the irregularity (anisotropy) of the grid, which is clearly reflected in the coarsening pattern. Since the grid is highly stretched in both directions x and y, near the bump, the coarsening algorithm is strongly influenced by that. For example, the coarsening pattern above the bump is nearly isotropic since the cell's aspect ratio is nearly unity, so the irregularity of the grid is not significant at this area. In addition, the discrete operator in this case relies upon a nine-point stencil which results in an "aggressive coarsening" (a similar coarsening pattern can be obtained by solving the Poisson equation with a nine-point stencil). However, as we move further away from the bump in the y-direction, the grid's aspect ratio gets bigger since the cells are stretched in the x-direction. Consequently, a given change in the derivatives along the x-direction would influence more strongly the new solution (depending on the grid's aspect ratio) than the same change in the derivative along the y-direction. The problem shows strong dependence in the x-direction, and little or no strong dependence in the y-direction. The same process

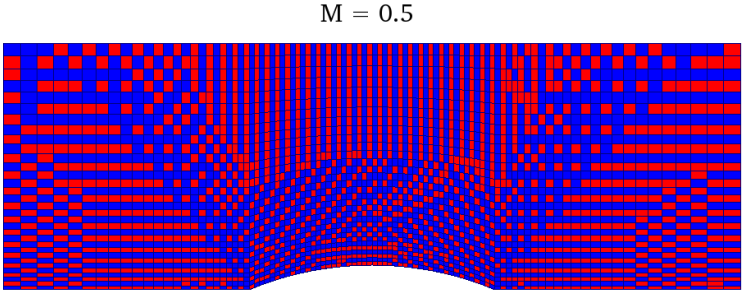
occurs in both sides of the bump, where the grid is coarsened in the y-direction — the direction of strong connections. This coarsening pattern for the rest of the flow conditions is essentially the same, accept for $M_\infty = 0.64$ where there is a slight disturbance of the regular coarsening above the bump, where the anisotropy is largest. It is important to mention here that a dynamic threshold is applied for this specific problem while in the rest of the subsonic cases a fixed threshold of $\varepsilon = 0.25$ results in good performance in terms of complexities and convergence properties.



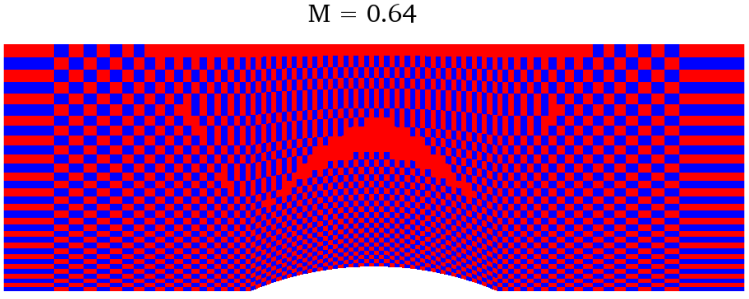
(a)



(b)



(c)



(d)

Figure 30: The finest and first coarse-level for flow through a channel with a bump. The mesh size is (96×32). The problem was solved with various Mach numbers: a) $M_\infty = 0.01$, b) $M_\infty = 0.25$, c) $M_\infty = 0.5$, d) $M_\infty = 0.64$. The blue point corresponds to F-point (fine-level) while the red point corresponds to C-point (coarse-level).

The discrete L_2-norm of the residual is shown in Table 8 for each V-cycle, in four different cases of M_∞. Note that for the first three cases and for all the grid sizes, the convergence factor is bounded well below 0.1. The residual norm decreases by a relatively the same factor with each V-cycle. This continues until it levels off after about 10 V-cycles near 10^{-13}, where round-off error is of the order of the residual norm itself. Due to the mathematical nature of the potential equation, the Mach number has a large impact on the convergence rate. When the Mach number increases so does the upwind bias of the discretization, and there is a decrease in the convergence rate as well. In addition, the effect of the nonlinearity and the existence of shock waves manifest themselves in the convergence properties for $M_\infty = 0.64$ where the first three cycles are slow to converge and then the residual reduction is stabilized on 0.1 for the remaining 7 V-cycles. It is rather clear that this slow convergence is only caused by the strong nonlinearity and the presence of a discontinuity. It takes $2-3$ "waste" cycles before the critical error components are sufficiently reduced by relaxations and the convergence becomes faster.

One possible way to improve the convergence rate at the beginning of the solution phase is by improving the initial condition. This can be achieved by the full–multigrid (FMG) approach, which is described in Section 8. The convergence of nonlinear iterations depends even more critically (compared to subsonic flow for instance) on a good initial condition. Typically, the better the initial condition used on the fine-level, the less significant is the effect of nonlinearity on the convergence and more effective the fine-level solver will be. The FMG approach was implemented in this problem for subsonic and supersonic flow as well. For subsonic flow it worked fairly well; the interpolation $I_{m+1}^m A^{m+1}$ is generally accurate enough to be treated by the fine-level relaxation. However, for supersonic flow, where an exceptionally strong nonlinearity exists the success is very limited. It is very important to mention that although the first three cycles are slow to converge, in the remaining V-cycles the convergence rate is lower than 0.1 without any local smoothing sweeps around the shock waves. To illustrate the residual reduction graphically, Figure 31 presents the convergence history of the residual (L_2 norm) versus the iterations number, for the four cases described above. We observe that for subsonic cases, $M_\infty = 0.1 - 0.5$ we need fewer than 10 V-cycles to reduce the L_2-norm of the residual by 10 orders of magnitude. As for the transonic case of $M_\infty = 0.64$, the convergence is somewhat slower here while about 12 cycles are required to reduce the residual by 10 orders of magnitude.

Table 8: The results of the AMG V-cycles applied to the flow through a channel with a bump with mesh size of 96×32. The discrete L_2-norm of the residual $\|R^m\|_2$ and the convergence rate C_f are presented after each V-cycle, for various Mach numbers. The sign (–) indicates that the level of discretization error has been reached.

	$M_\infty = 0.01$		$M_\infty = 0.2$		$M_\infty = 0.5$		$M_\infty = 0.64$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	3.236e-04	–	6.540e-03	–	5.587e-03	–	4.417e-03	–
1	1.320e-05	0.04	3.130e-04	0.05	1.135e-03	0.20	2.598e-03	0.58
2	3.763e-07	0.03	1.111e-05	0.04	1.536e-04	0.14	1.231e-03	0.47
3	1.295e-08	0.03	2.555e-07	0.02	2.071e-05	0.13	4.791e-04	0.39
4	4.234e-10	0.03	8.623e-09	0.03	1.484e-06	0.07	1.261e-04	0.26
5	1.397e-11	0.03	2.808e-10	0.03	8.585e-08	0.06	2.687e-05	0.21
6	4.404e-13	0.03	9.762e-12	0.03	3.655e-09	0.04	2.166e-06	0.08
7	1.324e-14	0.03	3.495e-13	0.04	1.492e-10	0.04	2.341e-07	0.11
8	4.154e-16	0.03	1.352e-14	0.04	9.411e-12	0.06	5.210e-08	0.22
9	1.326e-16	0.32	3.243e-15	0.24	6.416e-13	0.07	6.838e-09	0.13
10	1.317e-16	0.99	3.147e-15	0.97	3.537e-14	0.06	6.645e-10	0.10
11	1.319e-16	1.00	3.134e-15	1.00	8.233e-15	0.23	6.050e-11	0.09
12	1.332e-16	1.01	3.142e-15	1.00	7.900e-15	0.96	5.172e-12	0.09
13	–	–	–	–	–	–	3.934e-13	0.08
14	–	–	–	–	–	–	5.865e-14	0.15

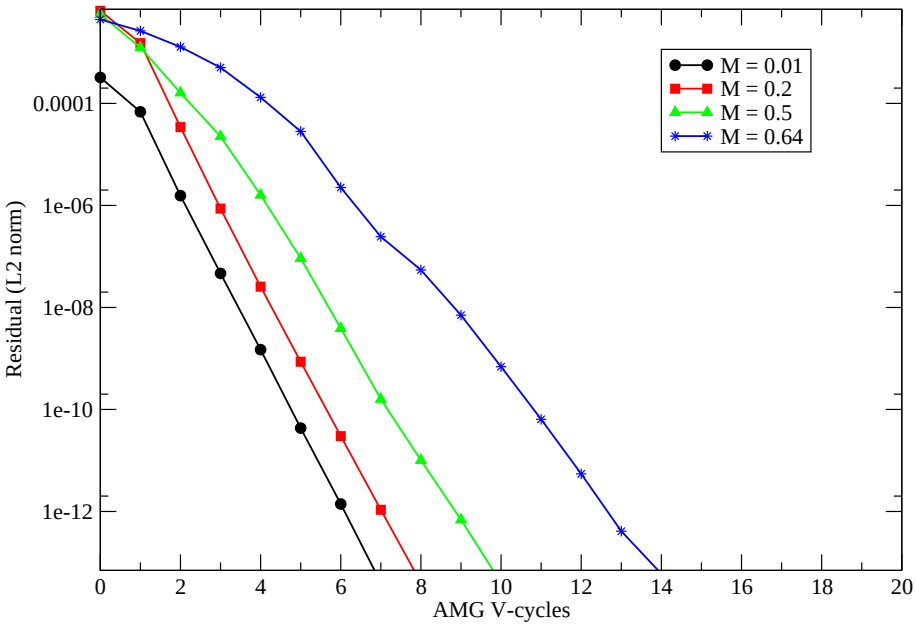


Figure 31: Convergence history of the discrete L_2-norm of the residual for various cases of Mach numbers.

Table 9: Grid complexity C_Ω and operator complexity C_L for four cases of Mach numbers.

Complexities	$M_\infty = 0.01$	$M_\infty = 0.2$	$M_\infty = 0.5$	$M_\infty = 0.64$
C_Ω	1.58	1.60	1.72	2.06
C_L	2.15	2.25	3.24	3.56

As mentioned, compared to isotropic problems, complexity is generally greater for anisotropic problems. The complexities can increase further for problems as discussed here where anisotropies are not aligned with the grid. Table 9 presents the grid and operator complexity for four test cases of Mach number. Summing the number of rows of all operators and dividing by the number of rows on the fine-level shows the grid complexity. For example, in the case of $M_\infty = 0.01$ the grid complexity is 1.58. Thus, we know that storage of the vector of unknowns (ϕ) and the right side requires 1.58 times the space required for the fine-level quantities. As the Mach number is increased the problem becomes strongly anisotropic and it results in an increased grid complexity. This is an expected behavior of the AMG algorithm since the memory (size of coarse-levels) requirement for the strongly anisotropic problem is greater than that for an isotropic problem. The reason is

that AMG is essentially performing one-dimensional coarsening in the direction of the strong connections. Therefore, when $M_\infty = 0.64$ a large "pocket" of supersonic flow is obtained above the bump, which is terminated by a shock wave. In this region, the anisotropy is largest. In this case the grid complexity is $C_\Omega = 2.06$.

The operator complexity slightly increases with the Mach number, and is above 2 for the four cases of Mach number. Two possible reasons for the increased operator complexity are the average stencil size and the coarsening process. The average stencil size is the average number of coefficients per row. For simplicity, let us look at the fine-level. When $M_\infty \ll 1$, the stencil size of the matrix A^* is large, although the matrix $\tilde{A}$ is diagonally dominant with $a_{i,j} \approx 1$ but the weights of the entries off-diagonal are definitely not zero (it is zero for $M_\infty = 0$). As the Mach number is increased so does the average stencil of A^*. It is possible to get very large stencil sizes on coarser levels (as will be discussed in Table 11). Large stencil size can lead to large operator complexity since various processes such as coarsening, interpolation, and relaxation, require that neighbors of neighbors are visited, which results in growth in the number of operations per cycle. The second reason for the increased operator complexity is the relatively large number of points on the coarse-levels, as previously explained. Furthermore, the second pass process, can also contribute to the relatively high complexities since F-points are replaced by C-points in order to satisfy the interpolation requirements.

For comparison , Table 10 presents the AMG performance for $M_\infty = 0.5$ while using a fixed threshold as a measure of strength between the matrix A^* entries, rather than dynamic threshold as presented above (in Table 8). This table includes the L_2-norm of the residual for 12 V-cycles. It is clearly seen that the alternative of a fixed threshold results in poor convergence properties and is also unstable with no upper limit. The grid and operator complexities obtained are $C_\Omega = 1.95$ and $C_L = 2.95$, respectively. However, the dynamic threshold approach was preferred for this specific problem, although the operator complexities were slightly increased.

Table 10: The results of the AMG V-cycles applied to the flow through a channel with a bump, with $M_\infty = 0.5$. The discrete L_2-norm of the residual $\|R^m\|_2$ and the convergence rate C_f are presented for the grid 96×32 after each V-cycle. A fixed threshold $\varepsilon = 0.25$ was applied during the coarsening process.

	$M_\infty = 0.5$	
V-cycle	$\|R^m\|_2$	C_f
0	1.724e-02	–
1	8.540e-03	0.50
2	1.076e-03	0.13
3	4.056e-05	0.04
4	1.622e-05	0.40
5	7.221e-06	0.45
6	1.705e-06	0.24
7	2.519e-07	0.15
8	7.893e-09	0.03
9	1.063e-08	1.35
10	3.988e-09	0.37
11	8.908e-10	0.22
12	1.195e-10	0.13
C_Ω	**1.95**	
C_L	**2.95**	

Let us examine the storage requirements for two extreme cases of $M_\infty = 0.64$ and $M_\infty = 0.1$. The problem was solved with 5 levels and the properties are presented in Table 11. The first coarse-level, A^{m+1}, has 1268 points, about one-fifth the number of points on the fine-level. This can be seen in Figure 30(a). Each succeeding coarse-level has approximately half the number of points as the preceding finer level. The coarse-level operator on the second level corresponds to a 29-point stencil. That is, although the reduction of points is substantial, the number of the nonzeros in the second- level matrix is still not far from the finest level. For $M_\infty = 0.65$ the situation is even more dramatic. In the last column it can be seen that the average number of nonzeros per row increases as the grids become coarser. However, the operator and grid complexities are slightly damaged. In addition, the first coarse-level has more actual nonzero coefficients than does the fine-level operator. This is relatively common for AMG and an example of this phenomenon is described by Briggs, Henson and McCormick [54]. The results were obtained by repeating the setup phase 6 times while applying two V-cycles between each update, resulting in a total of 12 V-cycles until the residual decreased to the desired level of $\tilde{}10^{-10}$.

A grid dependence study has been conducted to verify the independence behavior of the AMG algorithm on the grid resolution; two more grid levels have been used with (24×8) and (48×16) grid points in the axial and vertical directions. It is very important to mention that our main interest here is to verify the robustness of the code rather than achieving the greatest efficiency. Therefore, in each resolution, the AMG components (for instance, dynamic threshold, coarse-level parameter η, and second pass process) were "fixed"; that is, they were not

locally adjusted to particular requirements of a given resolution. These components contain exactly the values applied in the greatest resolution 96×32. The results of the L_2-norm of the residual $\|R^m\|_2$ and the convergence rate C_f are shown in Table 12. It can be clearly seen that the algorithm is scalable and does not depend on the problem size (the convergence factor is nearly constant for all the grids considered).

Table 11: Properties of the fine and coarse-level matrices for the AMG V-cycles applied to the flow through a channel with a bump.

	Number of rows		Number of nonzeros		Average entries per row	
Level	$M_\infty = 0.01$	$M_\infty = 0.64$	$M_\infty = 0.01$	$M_\infty = 0.64$	$M_\infty = 0.01$	$M_\infty = 0.64$
A^m	3072	3072	52760	52945	17.1	17.3
A^{m+1}	1238	1629	35511	56672	28.6	34.7
A^{m+2}	390	1000	18317	39568	47	39.5
A^{m+3}	126	564	5909	22743	46.5	40.3
A^{m+4}	41	296	1297	11305	31.6	38.1
A^{m+5}	–	148	–	5323	–	35.9

Table 12: The table shows the results of the AMG V-cycles applied to the flow through a nozzle. The second norm of the residual $\|R^m\|_2$ after each V-cycle and the convergence factor C_f are detailed for two mesh sizes, 25×9 and 49×17.

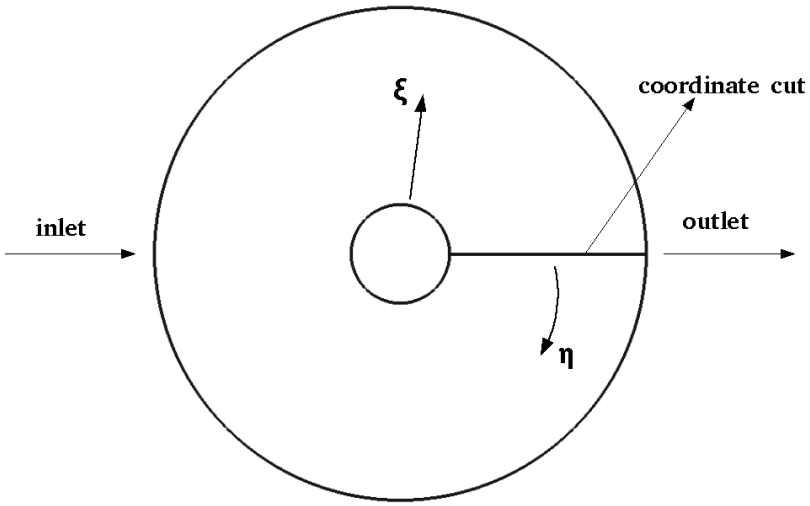
	$M_\infty = 0.01$				$M_\infty = 0.2$			
grid size $\longrightarrow$	25×9		49×17		25×9		49×17	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	5.664e-04	-	4.410e-04	-	1.152e-02	-	8.931e-03	-
1	2.072e-05	0.03	1.892e-05	0.04	5.169e-04	0.04	4.295e-04	0.05
2	7.042e-07	0.03	8.089e-07	0.04	2.221e-05	0.04	1.507e-05	0.04
3	2.495e-08	0.04	3.694e-08	0.05	9.302e-07	0.04	6.502e-07	0.04
4	8.551e-10	0.03	1.624e-09	0.04	3.626e-08	0.04	2.810e-08	0.04
5	2.759e-11	0.03	6.720e-11	0.04	1.304e-09	0.04	1.152e-09	0.04
6	8.339e-13	0.03	2.609e-12	0.04	4.319e-11	0.03	4.441e-11	0.04
7	2.371e-14	0.03	9.516e-14	0.04	1.330e-12	0.03	1.633e-12	0.04
8	6.407e-16	0.03	3.266e-15	0.03	4.090e-14	0.03	5.941e-14	0.04
9	4.256e-17	0.07	1.253e-16	0.04	1.764e-15	0.04	2.859e-15	0.05
10	4.026e-17	0.95	7.155e-17	0.57	8.928e-16	0.51	1.663e-15	0.58
11	4.043e-17	1.00	7.234e-17	1.01	9.090e-16	1.02	1.804e-15	1.08
C_Ω	**1.62**		**1.58**		**1.61**		**1.60**	
C_L	**1.87**		**2.04**		**1.88**		**2.12**	

	$M_\infty = 0.5$				$M_\infty = 0.64$			
grid size $\longrightarrow$	25×9		49×17		25×9		49×17	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	1.097e-03	–	3.203e-03	–	1.129e-03	–	3.748e-03	–
1	9.051e-05	0.08	1.188e-03	0.36	1.805e-04	0.16	8.262e-04	0.22
2	7.145e-06	0.08	1.732e-04	0.15	1.901e-05	0.11	2.990e-04	0.36
3	4.521e-07	0.06	2.294e-05	0.13	5.471e-07	0.03	5.535e-05	0.19
4	2.723e-08	0.06	2.025e-06	0.09	1.451e-08	0.03	6.562e-06	0.12
5	1.601e-09	0.06	1.634e-07	0.08	3.478e-10	0.02	6.091e-07	0.09
6	9.302e-11	0.06	1.252e-08	0.08	7.553e-12	0.02	4.411e-08	0.07
7	5.369e-12	0.06	9.381e-10	0.07	1.517e-13	0.02	2.941e-09	0.07
8	3.088e-13	0.06	6.945e-11	0.07	3.033e-15	0.02	1.882e-10	0.06
9	1.670e-14	0.05	5.109e-12	0.07	2.082e-15	0.69	1.185e-11	0.06
10	1.907e-15	0.11	3.743e-13	0.07	2.139e-15	1.03	7.417e-13	0.06
11	1.618e-15	0.85	2.634e-14	0.07	–	–	4.614e-14	0.06
12	1.947e-15	1.20	4.330e-15	0.16	–	–	4.684e-15	0.10
C_Ω	**1.70**		**1.75**		**2.03**		**2.02**	
C_L	**2.49**		**3.03**		**2.81**		**3.11**	

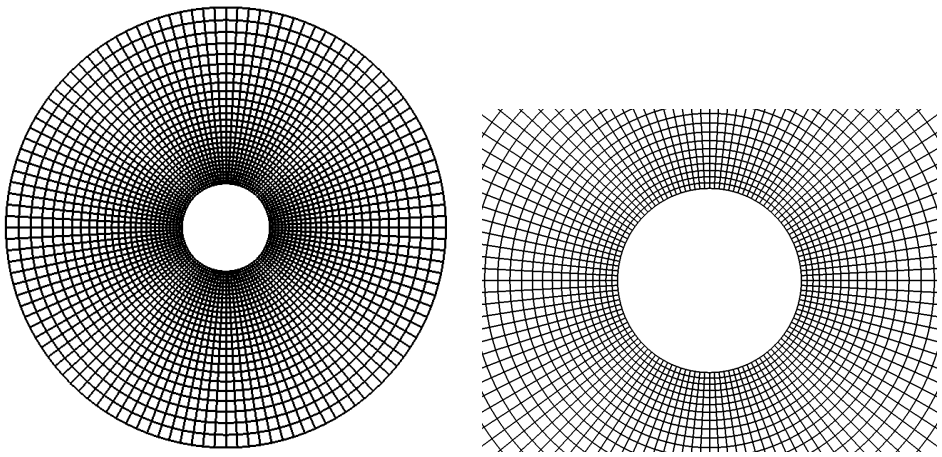
12.4 Flow around a circular cylinder

12.4.1 Computational domain and boundary conditions

We now consider a 2D cylinder placed in a uniform subsonic flow. The grid is sketched in Figure 32 and its generation is straightforward in polar coordinates, formed by circles and radial lines. The outer boundary is located far enough from the solid body where free undisturbed flow conditions are applied. In regions of strong flow variations, near the cylinder wall, the grid is refined. So the radial spacing of the circles increases from the solid body surface to the far field with a factor R, as defined by $\triangle r_i = r_{i+1} - r_i = r_i\,(R-1)$. The factor R is defined by the position of the outer boundary and the number of mesh points N_i in the radial direction: $r_{N_i} = R^{N_i-1}$. For example, selecting $N_i = 30$ points in the radial direction and an outer boundary at 5, we obtain $R = 5^{1/29} = 1.057$.



(a)



(b) (c)

Figure 32: An O-type mesh used for the circular cylinder flow test case. a) Extended mesh, b) Close-up of the mesh around the cylinder.

Along the ξ-line on the surface $i = 1$, the solid wall boundary condition was applied. The inlet and outlet flow boundary were located in $i = i_{max}$. The cut lies along the η-line ($j = 1, j_{max}$) where a periodic boundary condition is applied. The boundary conditions used are as follows:

- The farfield boundary ($\xi = max$) is five times radius lengths away from the cylinder, where the inflow and outflow boundary conditions were applied. A uniform flow (Neumann condition) was imposed in the x-direction at the inlet and outlet regions of the domain. In practice, this condition was attained by projecting the velocity vector normal to the cell's face.

- A solid wall boundary condition on the cylinder's surface. The normal velocity is zero since no mass penetrates the solid body. This condition is implemented according to Section 12.5.

- At $j = 1$ and $j = j_{max}$, along the cut, a coordinate cut boundary condition was applied. This is a line composed of grid points with different computational indices but the same physical location. The grid is folded such that it touches itself. The cut boundary condition is implemented by using ghost cells. The situation is sketched in Figure 33. The ghost cells coincide location-wise with the grid cells on the opposite side of the cut. Hence the values of the potential in the ghost cells are obtained directly from the opposite cells. All the fluxes across the faces of the boundary cell are evaluated exactly like in the interior field. The cut boundary is implemented by generating a complete control volume at the cut . Using the ghost cells (Numbers 0 and -1 in Figure 33), the fluxes were calculated in the same way as inside the domain. The value of the first ghost cell (number 0) is obtained directly from cell j_{max}. The value of the second ghost cell (number -1) is obtained from cell $j_{max} - 1$. We can write these conditions as follows:

$$\begin{aligned} \phi_{i,0} &= \phi_{i,j_{max}}, \\ \phi_{i,-1} &= \phi_{i,j_{max}-1}, \\ \phi_{i,j_{max}+1} &= \phi_{i,1}, \\ \phi_{i,j_{max}+2} &= \phi_{i,2}. \end{aligned} \tag{179}$$

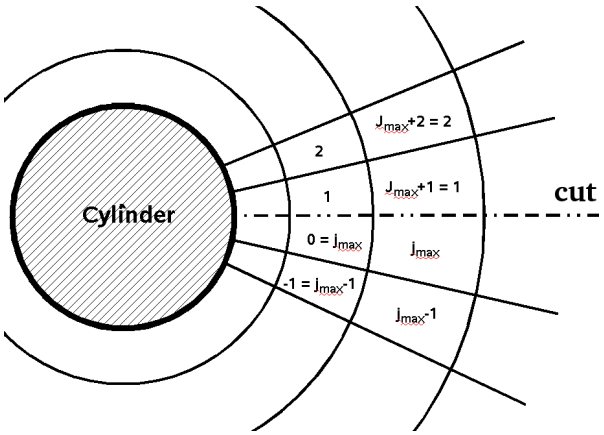


Figure 33: Coordinate cut boundary condition. Ghost cells are numbered as 0, -1, $(J_{max} + 1)$, and $(J_{max} + 2)$.

The results are presented for the fine-grid 28×120. The first number indicates the number of cells in the radial direction while the second number corresponds to the number of cells that are wrapped around the cylinder. The problem is solved for inlet Mach number values ranging from 0.01 to 0.41, and the results are presented in Figure 34. There are two stagnation points ($u = v = 0$) at $x = \pm a$, $y = 0$, at front and back ends of the cylinder. Within the low Mach number range, the solution does not greatly depend on the Mach number and practically

it is similar to the incompressible flow. When the Mach number is increased, the effect of the compressibility becomes more significant and for an inlet Mach number of 0.41, a sonic speed $M_\infty = 1$ was reached on the top and bottom of the cylinder surface (symmetric), followed by a supersonic region terminated by a shock wave. This can be seen for the case of an inlet velocity of $M_\infty = 0.41$.

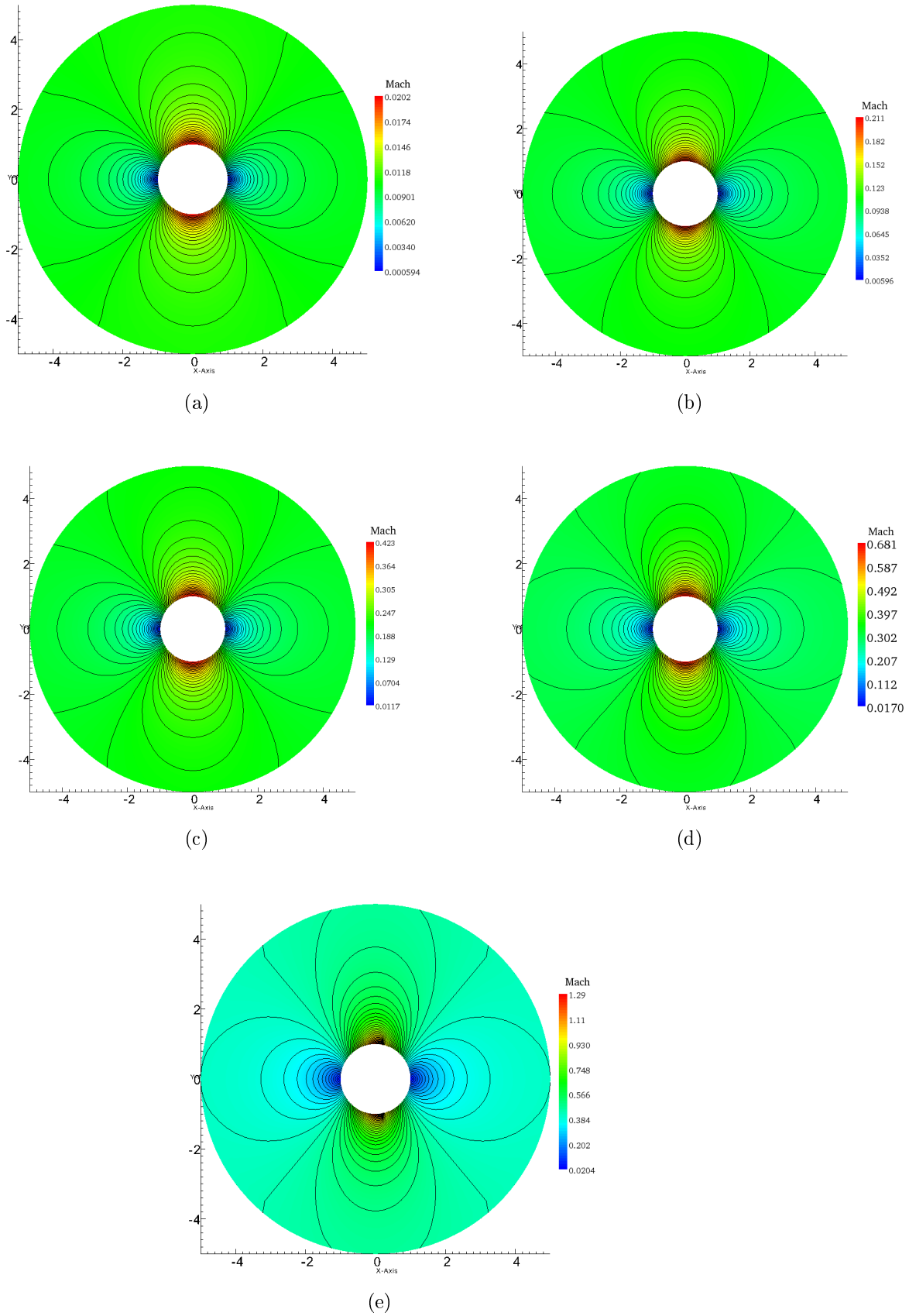
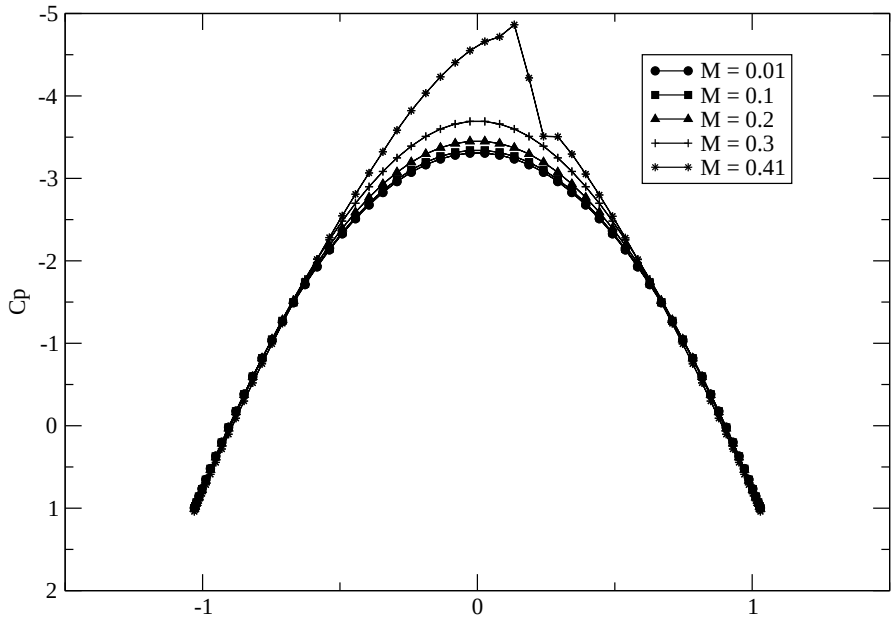


Figure 34: Distribution of velocity as computed on the (28×120) mesh, for an incident Mach number of: a.) $M_\infty = 0.01$, b) $M_\infty = 0.1$, c) $M_\infty = 0.2$, d) $M_\infty = 0.3$, e) $M_\infty = 0.41$. Observe the shock appearing at the top and bottom surfaces (case e).

The pressure distribution in a given flow pattern is of primary practical importance. Knowledge of the pressure distribution is necessary for the calculation of the forces and moments produced by the fluid on the solid boundaries. The pressure distribution at the cylinder surface can be found from Bernoulli's equation, $p + \frac{1}{2}\rho V^2 = const$. The pressure coefficient C_p is defined as:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2}, \tag{180}$$

where p_∞ represents the pressure at the farfield, and p is the pressure on the surface of the circular cylinder. A wide description of the pressure coefficient is described in the next section, which deals with the airfoil case study. The pressure coefficient distribution on the surface of the cylinder is plotted in Figure 35. Since the pressure distribution is symmetric about $x = 0$ and $y = 0$, there is no net force on the cylinder. Note that the pressure on most of the surface is less than p_∞ and the minimum value of C_p on the surface is -3 at $\theta = \pi/2$.



(a)

Figure 35: Surface pressure distribution along a circular cylinder with incident Mach number ranging from $M_\infty = 0.01$ to $M_\infty = 0.41$, using the (28×120) mesh.

12.4.2 AMG performance

The first coarse-level for each case described above is sketched in Figure 36. It is important to mention that for all the subsonic cases described above, $M_\infty \leq 0.5$, a fixed threshold of $\varepsilon = 0.25$ was applied, since much better results were observed, especially in terms of grid and operator complexity. However, the situation is vastly different in the transonic case, $M_\infty = 0.41$, where a dynamic threshold was applied in order to achieve convergence. Although this fast convergence is achieved at the expense of increased complexities (as presented in Table 14), the alternative of a fixed threshold $\varepsilon = 0.25$ in this specific case results in divergence. This specific point emphasizes the robustness of the improved coarsening process in the AMG algorithm. When an incompressible flow is addressed, the operator is nearly isotropic – a nine-point stencil. Therefore, as expected, the points that construct the coarse-level are distributed uniformly, as can be seen in Figure 36(a). This type of coarsening is typical for nine-point stencils with all connections being strong, yielding a grid complexity of ~ 1.6. If a dynamic threshold is applied for this problem it would result in less "aggressive" coarsening, and the complexities would deteriorate slightly. However, we use the dynamic threshold where we really need it, according to the requirements mentioned before.

As the velocity is increased the equation becomes anisotropic, and this uniform coarsening structure holds until the flow reaches supersonic speeds. This extreme anisotropy is characterized by a strong connection in the azimuth direction (μ - axis). There is a slight deviation from the uniform coarsening in the upper and lower parts of the cylinder, where the anisotropy of the problem is most significant. However, the coarsening pattern is essentially the same.

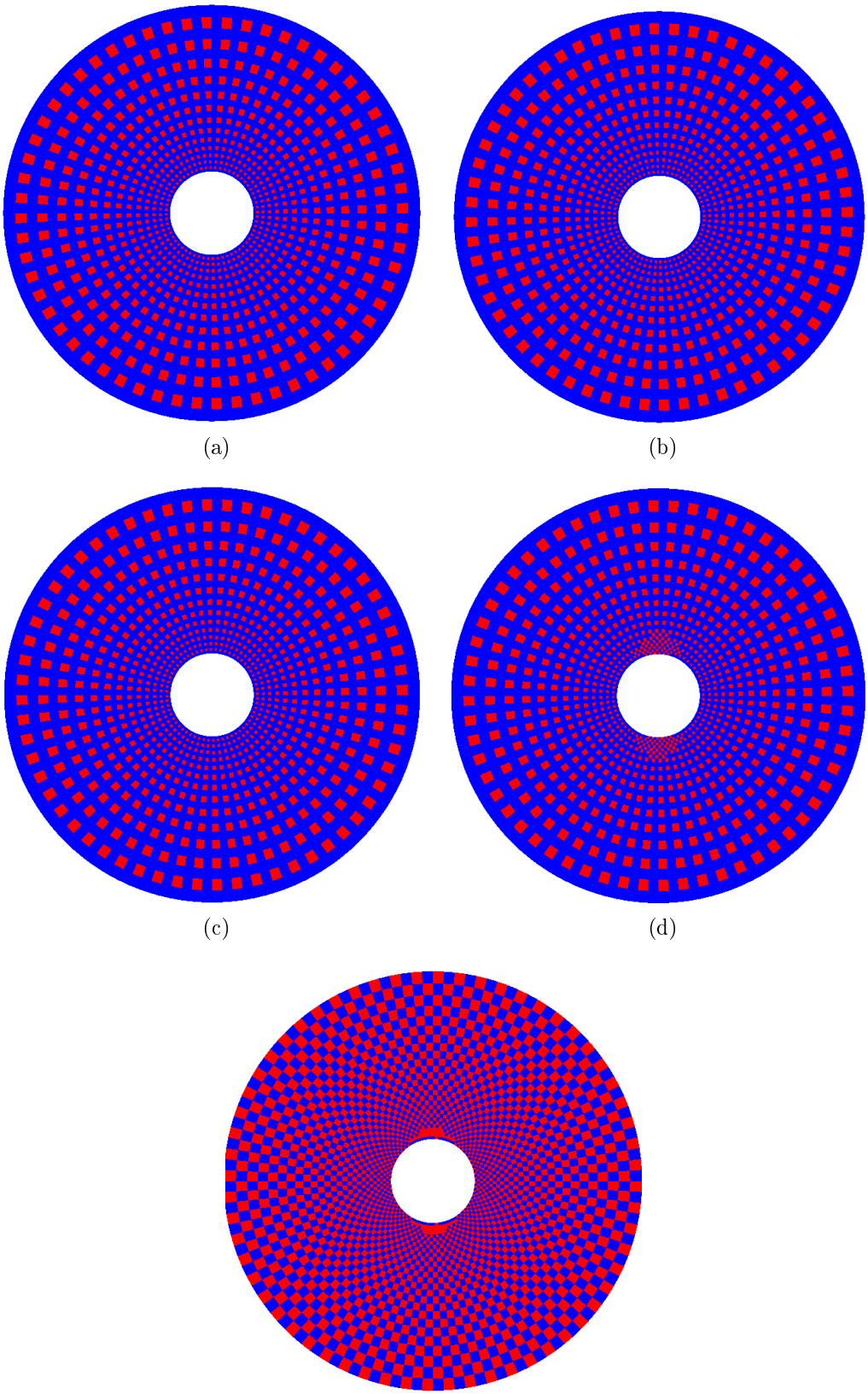


Figure 36: The finest and first coarse-level for mesh size (28×120). The red color corresponds to the C-point and the blue color corresponds to the F-point. The five free-stream velocities are as follows: a) $M_\infty = 0.01$, b) $M_\infty = 0.1$, c) $M_\infty = 0.2$, d) $M_\infty = 0.3$, e) $M_\infty = 0.41$.

Table 13 presents the discrete L_2-norm of the residual for each case of Mach number described above. The residual reduction is also sketched in Figure 37. The residual norm decreased rapidly for 10 to 12 V-cycles with the value in the corresponding $\|R^m\|_2$ column, while in the last few cycles a constant value was reached (asymptotic convergence rate). In the first three cases of $M_\infty = 0.01$, $M_\infty = 0.1$, and $M_\infty = 0.2$, the setup phase was performed, followed by 12 V-cycles. The convergence factors are bounded independent on the problem size. The convergence factor remains bounded independent of the problem size. In the case of $M_\infty = 0.3$, where the compressibility became significant, the results were obtained by repeating the setup phase 6 times while applying two V-cycles between each update. This results in a total of 12 V-cycles until the residual decreases to the desired level of $\tilde{}10^{-10}$. The transonic case provide to be a more difficult test for the algorithm. The convergence rate in the first three cycles is slightly damaged mainly due to the dominant nonlinearity. Although the first three cycles are relatively slow to converge, in the remaining V-cycles the convergence rate is less than 0.1 without any local smoothing sweeps around the shock waves.

The convergence histories for the five cases are depicted in Figure 37. The effect of the Mach number on the AMG performance is clearly shown. The case of $M_\infty = 0.41$ results in a supersonic flow regime that is terminated by a shock wave. The convergence is somewhat slower here. It requires nearly 12 V-cycles to decrease the 2-norm of the residual to a level of 10^{-10}, which is twice the number of cycles required in the case of $M_\infty = 0.01$.

The grid complexity and operator complexity are presented in Figure 14. It is clear that when the flow is subsonic the complexities are bounded. In the case of $M_\infty = 0.41$ the grid and operator complexity are too high, relative to our above requirements, while the reasons for the increased complexities are similar to those stated in the previous problem (channel with a bump). A possible way to improve the complexities is by aggressive coarsening. This approach was implemented but, as expected, convergence became considerably slower (above 0.3) and still approached an upper limit for the large mesh size. Aggressive coarsening not only causes the smoothing to be less effective but interpolation was also significantly less accurate.

Table 13: The results of AMG V-cycles applied to the flow around a circular cylinder. The L_2-norm of the residual after each V-cycle is presented for mesh size 28×120.

	$M_\infty = 0.01$		$M_\infty = 0.1$		$M_\infty = 0.2$		$M_\infty = 0.3$		$M_\infty = 0.41$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	7.489e-03	–	5.747e-02	–	0.115e-00	–	0.175e-00	–	7.542e-02	–
1	1.060e-03	0.14	1.235e-03	0.02	5.416e-03	0.05	1.390e-02	0.08	1.403e-02	0.18
2	4.306e-05	0.04	4.431e-05	0.04	7.122e-04	0.13	3.760e-03	0.27	6.565e-03	0.47
3	2.211e-06	0.05	1.886e-06	0.04	1.024e-04	0.14	3.615e-04	0.10	1.171e-03	0.18
4	1.201e-07	0.05	8.440e-08	0.04	4.352e-06	0.04	2.957e-05	0.08	6.378e-04	0.54
5	6.850e-09	0.06	3.884e-09	0.05	1.855e-07	0.04	1.914e-06	0.06	5.716e-05	0.09
6	4.048e-10	0.06	1.816e-10	0.05	7.848e-09	0.04	1.151e-07	0.06	5.244e-06	0.09
7	2.457e-11	0.06	8.533e-12	0.05	3.283e-10	0.04	6.721e-09	0.06	3.419e-07	0.07
8	1.508e-12	0.06	3.998e-13	0.05	1.364e-11	0.04	3.874e-10	0.06	2.524e-08	0.07
9	9.300e-14	0.06	1.874e-14	0.05	5.636e-13	0.04	2.222e-11	0.06	1.760e-09	0.07
10	5.773e-15	0.06	1.893e-15	0.10	2.308e-14	0.04	1.272e-12	0.06	1.203e-10	0.07
11	4.067e-16	0.07	1.736e-15	0.92	3.620e-15	0.16	7.290e-14	0.06	8.136e-12	0.07
12	1.798e-16	0.44	1.700e-15	0.98	3.367e-15	0.93	6.622e-15	0.09	5.484e-13	0.07

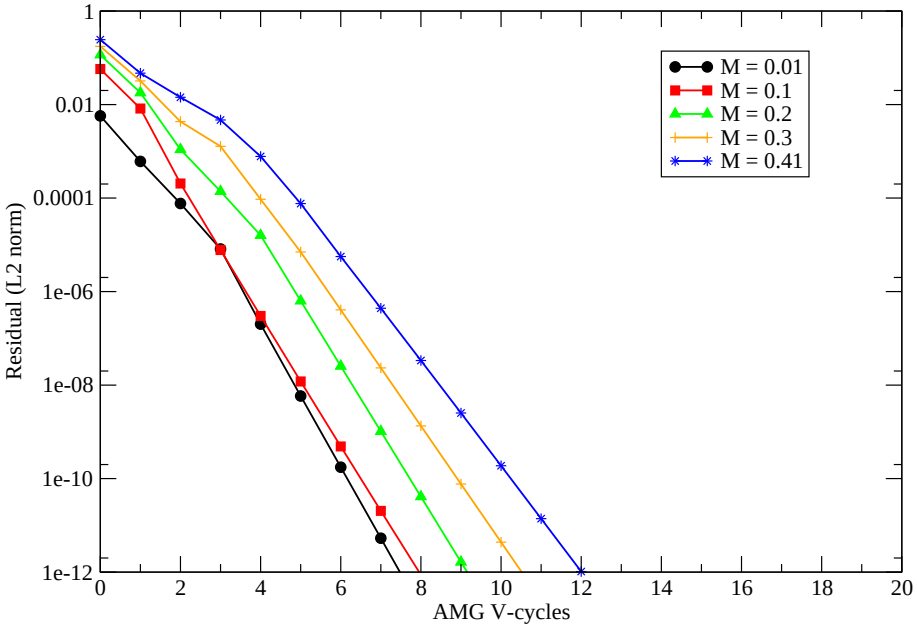


Figure 37: Convergence histories.

Table 14: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

complexities	$M_\infty = 0.01$	$M_\infty = 0.1$	$M_\infty = 0.2$	$M_\infty = 0.3$	$M_\infty = 0.41$
C_Ω	1.31	1.33	1.33	1.33	1.98
C_L	1.82	1.867	2.26	2.18	3.09

A useful tool to examine how AMG performs on the problem is to analyze the coarsening statistics. Table 15 shows the average number of rows and the number of nonzeros for cases of $M_\infty = 0.01$ and $M_\infty = 0.41$. In $M_\infty = 0.41$ the problem was solved with 6 levels. The initial coarse-level, A^{m+1}, has 1700 points, exactly half the number of points on the fine-level. This can be seen in the color scheme described in Figure 36(a). The next succeeding coarse-levels contain approximately half the number of points as the preceding finer level. Subsequent coarsening then becomes slightly faster because the coarse-level, which was produced by direct approximation (described in Section 8), becomes larger on coarser levels. The reason for this relatively slow coarsening is due to the dynamic threshold as a measure of strength between the matrix variables. The alternative of using a fixed threshold, particularly in this transonic case, would result in a significantly slower convergence. In the last column it can be seen that the average number of entries per row increases as the grids become coarser. However, the operator and grid complexities are not damaged. Table 15 presents also the coarse-levels for $M = 0.01$. In this case a fixed threshold is applied. It is clearly seen that the AMG coarsening is much faster, with only 4 coarse-levels. The initial coarse-level, A^{m+1}, has 780 points, less than one-fourth the number of points on the fine-level. This coarsening pattern is essentially the same for the next coarse-levels. This example exactly introduces the reason why a fixed threshold is preferred in the subsonic case, where the equation is elliptic, instead of using a dynamic threshold. More in this context, the reason why a relatively large stencil of 16 points is obtained on the fine-level is due to the weights of the operator $\tilde{L}$, although they are very small and "far away" from the diagonal, their value is nonzero.

This problem was solved for two grids consisting of 14×60 and 7×30 points; the results are presented in Table 16. The convergence factors are bounded independent of the problem size. The complexities of the AMG cycles remain nicely bounded independent of the problem size. The grid complexity and operator complexity for grid 7×30 are $C_\Omega = 1.72$ and $C_L = 2.13$, respectively, and for the case of grid 14×60 the results are $C_\Omega = 1.31$ and $C_L = 1.72$.

Table 15: Results of the AMG V-cycles applied to the flow around a circular cylinder.

	Number of rows		Number of nonzeros		Average entries per row	
Level	$M_\infty = 0.01$	$M_\infty = 0.41$	$M_\infty = 0.01$	$M_\infty = 0.41$	$M_\infty = 0.01$	$M_\infty = 0.41$
A^m	3360	3360	55924	58260	16.6	17.3
A^{m+1}	780	1700	35460	57868	45.4	34
A^{m+2}	180	884	8940	31141	49.6	35.2
A^{m+3}	45	456	1605	17891	35.6	39.2
A^{m+4}	–	238	–	9749	–	40.9
A^{m+5}	–	126	–	5203	–	41.2

Table 16: The results of the AMG V-cycles applied to the flow around a circular cylinder. The second norm of the residual after each V-cycle $\|R^m\|_2$, convergence factor C_f, grid complexity C_Ω, and operator complexity C_L are presented for grid sizes 7×30 and 14×60.

	$M_\infty = 0.2$				$M_\infty = 0.41$			
grid size $\longrightarrow$	7×30		14×60		7×30		14×60	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	0.261e-00	–	0.182e-00	–	3.409e-02	–	2.323e-02	–
1	1.558e-02	0.06	1.093e-02	0.06	2.303e-03	0.06	2.395e-03	0.1
2	1.029e-03	0.07	1.107e-03	0.10	2.179e-04	0.09	5.023e-04	0.21
3	7.318e-05	0.07	1.230e-04	0.10	9.027e-06	0.04	3.197e-05	0.06
4	2.343e-06	0.03	4.268e-06	0.03	2.480e-07	0.03	1.644e-06	0.05
5	7.183e-08	0.03	1.489e-07	0.03	7.155e-09	0.03	8.751e-08	0.05
6	2.183e-09	0.03	5.264e-09	0.04	2.145e-10	0.03	5.100e-09	0.06
7	6.725e-11	0.03	1.884e-10	0.04	6.623e-12	0.03	3.431e-10	0.07
8	2.158e-12	0.03	7.065e-12	0.04	2.082e-13	0.03	2.452e-11	0.07
9	7.410e-14	0.03	2.807e-13	0.04	7.384e-15	0.04	1.770e-12	0.07
10	3.964e-15	0.05	1.181e-14	0.04	2.571e-15	0.35	1.274e-13	0.07
11	2.165e-15	0.55	1.853e-15	0.16	2.397e-15	0.93	9.721e-15	0.08
12	2.754e-15	1.27	2.034e-15	1.10	2.592e-15	1.08	3.795e-15	0.39
C_Ω	**1.72**		**1.31**		**1.88**		**2.02**	
C_L	**2.13**		**1.72**		**2.45**		**2.92**	

12.5 Flow around a circular cylinder with circulation

A rotating circular cylinder placed in a free-stream generates a fluid motion that has been the subject of research from the point of view of numerical simulations, experiments, and theoretical analysis. This flow field is equivalent to the combination of flow past a cylinder and a vortex. This type of fluid motion is of major importance in aerodynamics, since fundamental aspects of the flow past an airfoil

(which is the next numerical experiment) can be enlightened through conformal transformation from the flow past a rotating cylinder, under the appropriate simplifications. Two aspects have drawn attention from researchers with respect to flow past a rotating cylinder. The first aspect is that the rotation action is able to suppress the separation of the boundary layer around the cylinder. The second aspect is the lift generated on the cylinder by the surrounding fluid, also known as the Magnus effect [16, 18].

In this problem the same O-type mesh is used, and the only difference is in the boundary conditions. Exactly as in the case with the fixed cylinder, there is an outer boundary where an inflow and outflow are applied. The circulation around the cylinder is applied by using the cut, emanating from the body to the farfield, where a jump in the potential is allowed. Thus, the cut can be interpreted as a periodic boundary with conditions, for any point on the cut,

$$\begin{aligned} \phi_{i,0} &= \phi_{i,jmax} + \Gamma, \\ \phi_{i,-1} &= \phi_{i,jmax-1} + \Gamma, \\ \phi_{i,jmax+1} &= \phi_{i,1} - \Gamma, \\ \phi_{i,jmax+2} &= \phi_{i,2} - \Gamma. \end{aligned} \tag{181}$$

The results are presented for the fine-level that includes 3360 grid points. The algorithm was tested by several flow conditions as follows:

1. $M_\infty = 0$, $\Gamma = 0.01$
2. $M_\infty = 0.1$, $\Gamma = 0.01$
3. $M_\infty = 0.1$, $\Gamma = 0.05$
4. $M_\infty = 0.1$, $\Gamma = 0.1$
5. $M_\infty = 0.4$, $\Gamma = 0.1$

The first step was to validate the problem setup, the choice of the boundary conditions, and the mesh attributes. This accomplished by applying a circulation of $\Gamma = 0.01$ with a nearly zero free-stream ($M_\infty \approx 0$, case 1). The sign of the circulation imposes a flow in the clockwise direction. The Mach isolines for this case are plotted in Figure 38. Since a zero free-stream is applied, the Mach contours are simply symmetric contours around the cylinder. The problem was solved for inlet Mach number values ranging from 0.01 to 0.4, and the results are presented in Figure 39. Only the fourth case results in a supersonic flow regime. Notice that the stagnation point lies above the cylinder, in the region where the direction of the free-stream opposes the circulation. As the flow's speed at the surface of the cylinder increases, the region of close isolines around the cylinder extends far from the wall and, as a consequence, the stagnation point moves upward.

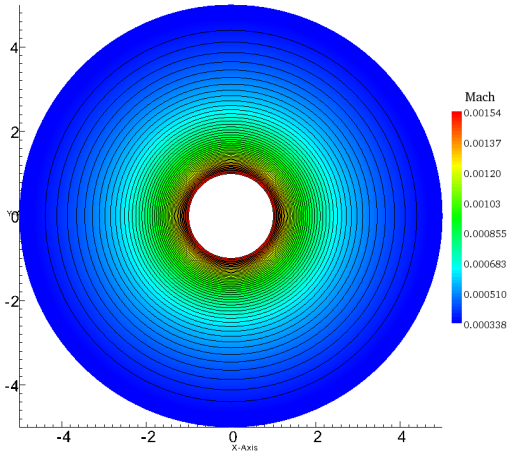


Figure 38: Flow over a cylinder having a circulation of $\Gamma = 0.01$. Contours are spaced for equal increments of 10^{-5}.

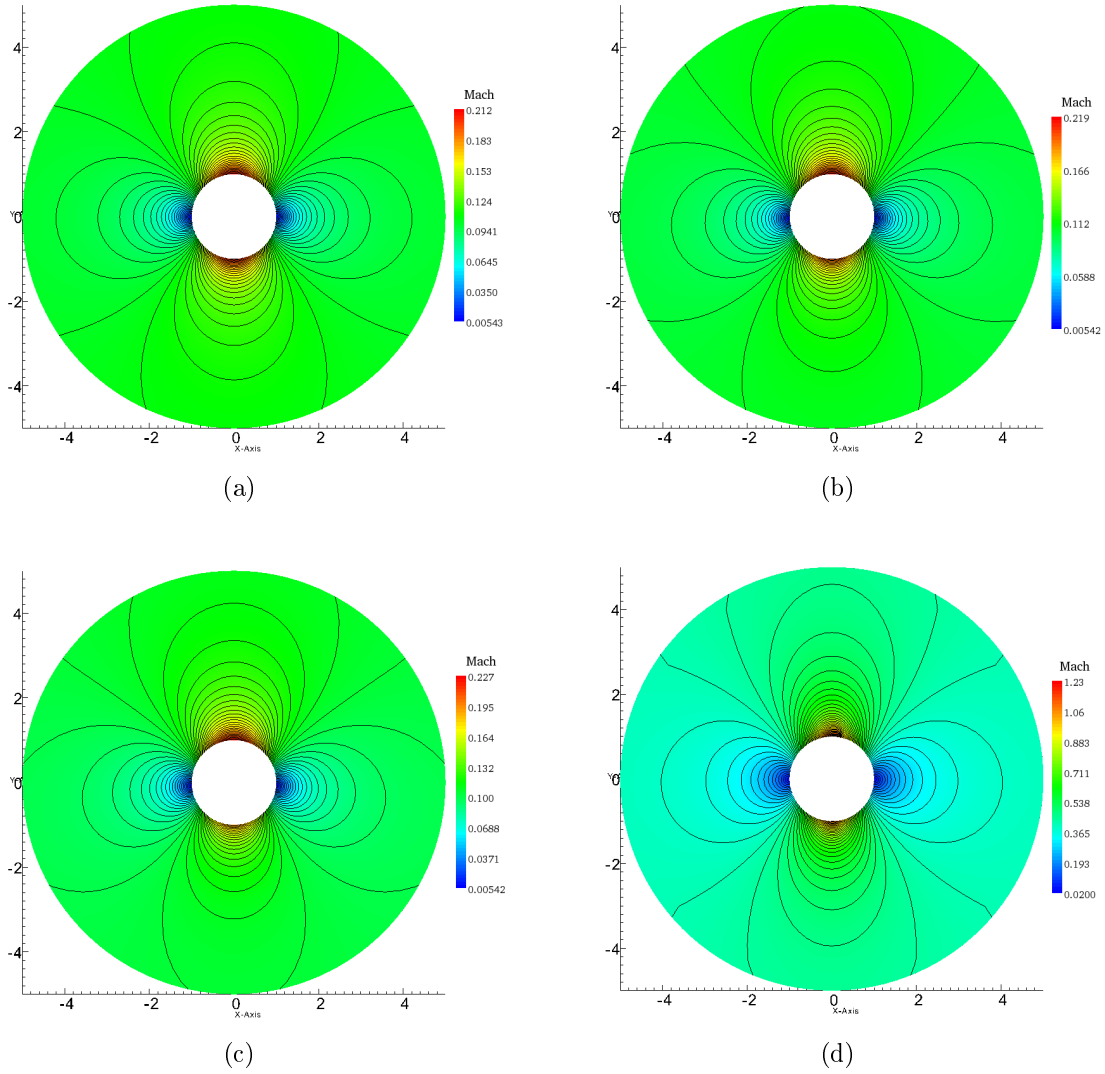


Figure 39: Distribution of velocity as computed on the (28×120) mesh, for the following flow characteristics: a) $M_\infty = 0.1$, $\Gamma = 0.01$, b)$M_\infty = 0.1$, $\Gamma = 0.05$, c)$M_\infty = 0.1$, $\Gamma = 0.1$, d) $M_\infty = 0.4$, $\Gamma = 0.1$. Observe the shock appearing at $M_\infty = 0.41$.

It was verified in these simulations that the velocity becomes close to the free-stream velocity along the outer boundary of the domain. The pressure coefficient C_p distribution on the surface of the cylinder is plotted in Figure 40. It is clear that when the flow is subsonic in the entire domain there is no significant difference in the C_p profiles. From Figure 40 we can see the pressure difference between the lower and the upper part of the cylinder. This is the lift generated on the cylinder by the surrounding fluid, also known as Magnus effect [18]. It is clear that the lift is increased with the free-stream. As the flow becomes supersonic on the upper part of the cylinder's surface, a shock wave appears. It is clearly seen in the sharp decrease in the pressure coefficient. The tendencies described by the classical irrotational theory (see, e.g., Batchelor [18]) for the pressure coefficient distribution are followed by the numerical solution obtained in this investigation.

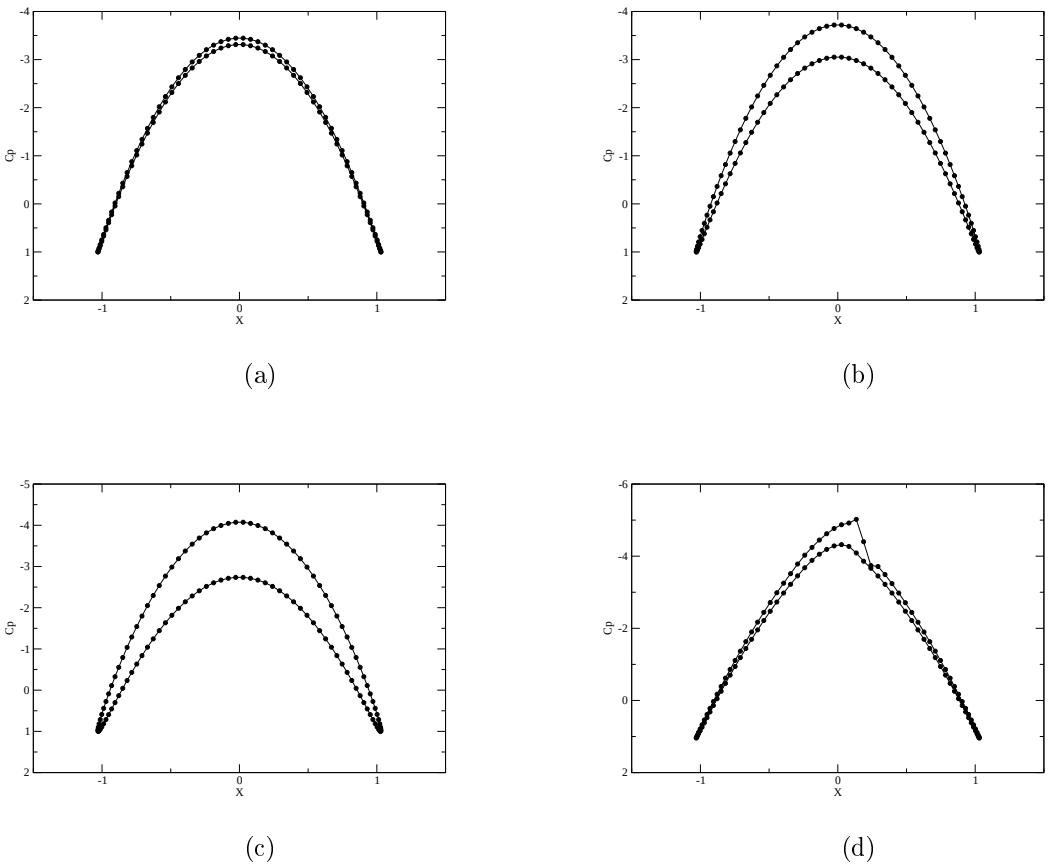
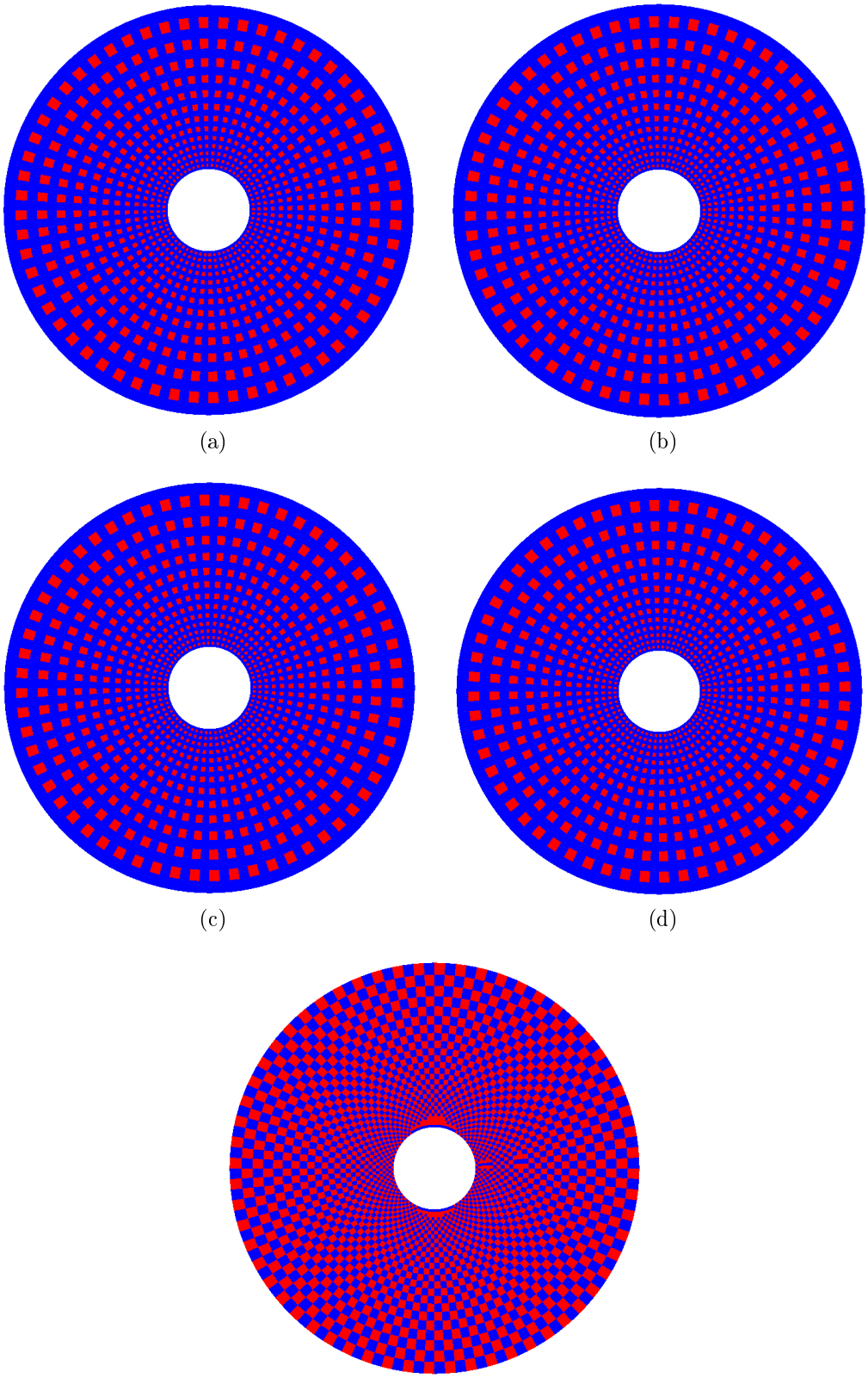


Figure 40: The distribution of pressure coefficient C_p by using (28×120)mesh. The cases are as follows: a) $M_\infty = 0.1$, $\Gamma = 0.01$, b)$M_\infty = 0.1$, $\Gamma = 0.05$, c)$M_\infty = 0.1$, $\Gamma = 0.1$, d) $M_\infty = 0.4$, $\Gamma = 0.1$.

12.5.1 AMG performance

The first coarse-level for each case described above is sketched in Figure 41. When the flow is subsonic, $M_\infty < 0.5$, it is characterized by a nearly isotropic operator and the C-points the construct the coarse-level are distributed uniformly, as can be seen in Figure 41a. As the free-stream is increased to the supersonic case, as mentioned above, there is a slight disturbance of the uniform coarsening pattern where the anisotropy of the operator is largest (see Figure 41(e)).



(a) (b) (c) (d) (e)

Figure 41: The finest and first coarse-level. The red color corresponds to the C-points while the blue color corresponds to the F-points. The mesh size is (28×120). The five flow conditions are as follows: a) case 1, b) case 2, c) case 3, d) case 4, e) case 5.

The convergence of the method is summarized in Table 17 for the five cases under consideration. Using the standard L_2-norm we see that after 12 cycles, the residual reached 10^{-15} and the AMG converged rapidly for the cylinder with circulation as for the previous model problem, where we saw a convergence rate lower than an order of magnitude for all the cases that were previously introduced. The fastest cycle, $M_\infty = 0.1$ and $\Gamma = 0.05$, needs 11 steps to reduce the residual by ten orders of magnitude. In cases $1-4$ no updates of the matrix $A^*$and the restriction and interpolation operators were needed. In case 5, the nonlinearity is dominant, which required to execution of the setup phase 6 times with 2 V-cycles in between, resulting in 12 V-cycles overall to reach convergence.

The complexities for these five cases are presented in Table 18. The operator complexity is below 2 for cases $1-4$ where the flow is subsonic in the entire field. Case 5 is characterized by a transonic flow on the upper part of the cylinder, and the grid and operator complexities are increased to 1.97 and 3.0, respectively. Two possible reasons for the increased operator complexity are the average stencil size and the coarsening process. The average stencil size is increased due to the existence of the operator $\tilde{L}$ when the velocity approaches supersonic values. The second reason for the increased operator complexity is the relatively large number of points on the fine-levels, which is a result of the second-pass process, when F-points are replaced by C-points in order to satisfy the interpolation requirements.

Table 17: The results of AMG V-cycles applied to the flow around a circular cylinder with circulation. The second norm of the residual after each V-cycle is presented for mesh size 28×120.

	case 1		case 2		case 3		case 4		case 5	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	8.293e-02	–	0.112e-00	–	0.396e-00	–	0.702e-00	–	7.431e-02	–
1	2.141e-03	0.02	9.882e-03	0.04	1.594e-02	0.04	3.462e-02	0.05	1.208e-02	0.16
2	5.941e-05	0.03	3.965e-04	0.04	6.093e-04	0.04	1.344e-03	0.04	1.774e-03	0.15
3	1.782e-06	0.03	2.233e-05	0.06	2.981e-05	0.05	5.913e-05	0.04	2.106e-04	0.12
4	5.783e-08	0.03	1.429e-06	0.06	1.706e-06	0.06	2.911e-06	0.05	1.988e-05	0.09
5	2.036e-09	0.04	9.648e-08	0.07	1.078e-07	0.06	1.583e-07	0.05	1.805e-06	0.09
6	7.814e-11	0.04	6.594e-09	0.07	7.130e-09	0.07	9.387e-09	0.06	1.606e-07	0.09
7	3.273e-12	0.04	4.499e-10	0.07	4.782e-10	0.07	5.885e-10	0.06	1.414e-08	0.09
8	1.504e-13	0.05	3.047e-11	0.07	3.207e-11	0.07	3.795e-11	0.06	1.235e-09	0.09
9	7.187e-15	0.05	2.048e-12	0.07	2.144e-12	0.07	2.481e-12	0.07	1.073e-10	0.09
10	3.482e-16	0.05	1.369e-13	0.07	1.428e-13	0.07	1.635e-13	0.07	9.288e-12	0.09
11	2.997e-17	0.09	9.293e-15	0.07	9.584e-15	0.07	1.094e-14	0.07	8.013e-13	0.09
12	2.309e-17	0.77	2.082e-15	0.22	1.995e-15	0.21	2.660e-15	0.24	6.890e-14	0.09

Table 18: Grid complexity C_Ω and operator complexity C_L for five cases of Mach number and circulation magnitude.

Complexities	case 1	case 2	case 3	case 4	case 5
C_Ω	1.31	1.31	1.31	1.31	1.97
C_L	1.82	1.82	1.82	1.82	3.016

12.6 Symmetric airfoil – NACA-0012

In the field of fluid dynamics, an area of significant practical importance is the study of airfoils. An airfoil refers to the cross-sectional shape of an object designed to generates lift when moving through a fluid. The airfoil generate lift by diverting the motion of fluid flowing over its surface in a downward direction, which results in an upward reaction force.

12.6.1 Airfoil characterization

Before dealing with the fluid flow around the airfoil let us define the airfoil geometrically. The main parameters of an airfoil are labeled in the diagram presented in Figure 42. Specifically we are interested in the angle of attack α, the chord length, and the mean camber line. The chord is a straight line and typically used to measure the airfoil length. The mean camber line is a curve that connects halfway between the upper and lower surfaces, and is used to measure the airfoil curvature. Airfoil shapes are commonly characterized with a numbering system originally defined by the National Advisory Committee for Aeronautics (NACA). This characterizing system defines airfoil shapes with a series of digits corresponding to nondimensional airfoil properties. In this work we modeled a 4 digit airfoil to simplify the geometry of the airfoil we wish to analyze. The first number describes the maximum camber as a percentage of the chord length. The second digit indicates the position of the maximum camber in tenths of the chord. The last two digits provide the maximum thickness of the airfoil as a percent of the chord length.

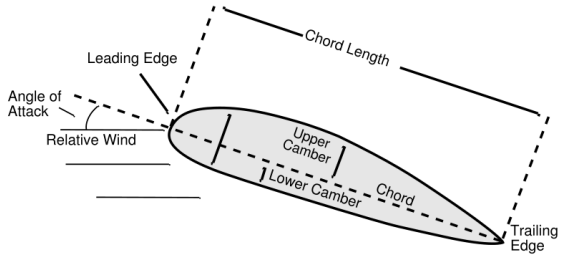


Figure 42: Diagram of an airfoil with key parameters labeled.

12.6.2 Problem definition

A grid convergence study was done in order to determine the appropriate grid dimensions in terms of the distance between the airfoil and farfield, grid resolution, etc. Four different grids were tested. They are: (128×48) , (64×24), (32×12). The first number corresponds to the number of cells that wrap around the airfoil surface, and the second dimension is the number of cells normal to airfoil's surface. In this case an O-topology mesh is generated by applying the algorithm of Donald Hawken [95], as is sketched in Figure 44(a) . A close-up of the grid near the airfoil is shown in Figure 44(b). The diagram of the airfoil is sketched in Figure 43. We can see from the rendering of the O-topology that the aerodynamic body is enclosed by one family of grid lines ($\eta = const.$). The second family of grid lines ($\xi = const.$) is spanned in the radial direction between the body and the farfield. The complete boundary line $\eta = 1$ represents the contour of the body, from a to b. The coordinate cut is defined by the boundaries $\xi = 1$ (nodes a-c) and $\xi = max$ (nodes b-d) in the computational space. The coordinates of the NACA-0012 profile are given in Table 19.

Table 19: Coordinates of the NACA-0012 airfoil.

$x, [m]$	$y, [m]$	$x, [m]$	$y, [m]$
0.5	0	-0.49685	0.00971
0.49814	-0.00032	-0.48741	0.01891
0.49248	-0.00108	-0.47176	0.0275
0.48192	-0.00253	-0.44959	0.03549
0.46487	-0.00488	-0.42028	0.04278
0.43958	-0.00826	-0.38319	0.04914
0.40455	-0.01278	-0.33776	0.05427
0.35889	-0.0184	-0.28373	0.05779
0.3026	-0.02492	-0.22135	0.05941
0.2367	-0.03201	-0.15147	0.05893
0.16313	-0.03923	-0.07563	0.05636
0.08454	-0.04605	0.00395	0.05194
0.00395	-0.05194	0.08454	0.04605
-0.07563	-0.05636	0.16313	0.03923
-0.15147	-0.05893	0.2367	0.03201
-0.22135	-0.05941	0.3026	0.02492
-0.28373	-0.05779	0.35889	0.0184
-0.33776	-0.05427	0.40455	0.01278
-0.38319	-0.04914	0.43958	0.00826
-0.42028	-0.04278	0.46487	0.00488
-0.44959	-0.03549	0.48192	0.00253
-0.47176	-0.0275	0.49248	0.00108
-0.48741	-0.01891	0.49814	0.00032
-0.49685	-0.00971	0.5	0
-0.5	0	–	–

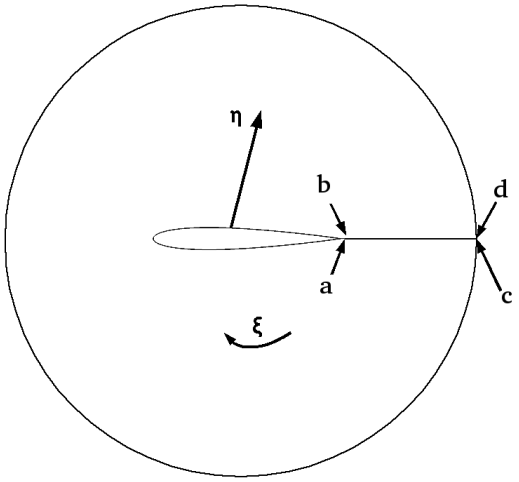
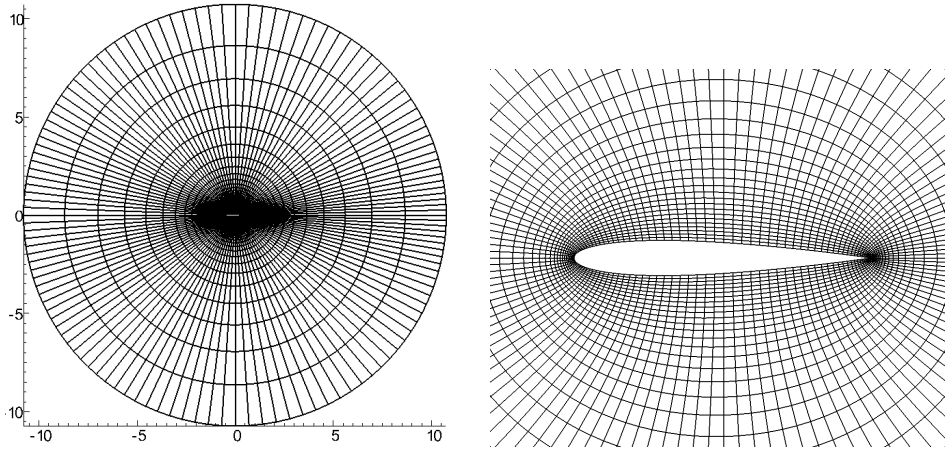


Figure 43: Diagram of the airfoil in O-type topology.



(a) (b)

Figure 44: Structured curvilinear body-fitted grid of the O-type. a) Mesh used for the NACA-0012 airfoil flow case study. b) A close-up of the grid near the airfoil.

Definitions of the domain boundaries and associated boundary conditions are as follows. The farfield boundary ($\eta = max$) is ten chord lengths away from the airfoil, where the inflow and outflow boundary conditions were imposed. On the surface of the airfoil the tangency condition was enforced. The cut boundary is implemented by generating a complete control volume at the cut. The details of how to apply the periodic boundary condition along the cut are exactly as described in the problem of flow around a circular cylinder (see Section 12). In this context, it is important to mention that all the ghost cells along the cut are constructed so that they perfectly fit the geometry of the cells that they intend to replace.

Since the potential assumption implies the flow is irrotational, the local vorticity production is zero. When solving a flow problem involving an airoil that produces lift, there is a finite circulation Γ around the airfoil. In order to obtain lift, this circulation must be imposed. Its magnitude is obtained by applying the Kutta-Joukowski condition, which states that the circulation value is such that the stagnation point is located at the trailing edge. The details of how to implement the Kutta-Joukowski condition are described below.

12.6.3 The Kutta-Joukowski condition

Consider an airfoil at an angle of attack, producing a nonzero lift. The following line integral over any contour enclosing such an airfoil will produce a nonzero result, known as the circulation around an airfoil Γ,

$$\oint_{\partial s} \overrightarrow{V} \cdot \overrightarrow{dS} = \Gamma. \qquad around\ airfoil. \tag{182}$$

In the above integral, dS is an infinitesimal line segment vector, tangential to the contour. In the case of potential flow, then, using the Stokes's theorem, the above line integral may be shown to be equivalent to the following area integral:

$$\oint \overrightarrow{_{\partial s} V} \cdot \overrightarrow{dS} = \iint_{\Omega} \left(\nabla \times \overrightarrow{V} \right) \cdot \overrightarrow{n} \cdot d\Omega = 0. \tag{183}$$

The area integral is zero because the curl of the velocity vector is zero in a potential flow. Now let us link the above integral to the velocity potential. The integrand may be written as follows:

$$\overrightarrow{V} \cdot dS = \overrightarrow{\nabla} \phi \cdot \overrightarrow{dS} = \left(\phi_x \hat{i} + \phi_y \hat{j} \right) \left(dx \hat{i} + dy \hat{j} \right) = \frac{\partial \phi}{\partial x} dx + \frac{\partial \phi}{\partial y} dy = d\phi. \tag{184}$$

Thus, the circulation Γ in Eq. (184) is related to the jump in the velocity potential:

$$\oint_{\partial s} \overrightarrow{V} \cdot \overrightarrow{dS} = d\phi \qquad around\ airfoil. \tag{185}$$

The above result means that somewhere in the $x - y$ plane, the potential function must experience an abrupt jump in its value by a quantity equal to the circulation Γ. In our analysis, we locate the disturbance potential jump along a cut that starts at the airfoil trailing edge, and ends at the downstream infinity. Figure 45 gives an illustration of the trailing edge in 2D airfoil problem. Consider a typical cell $(i, 1)$ above the cut, downstream of airfoil trailing edge. The circulation is first computed as,

$$\Gamma = \phi_{TE,top} - \phi_{TE,bottom}, \tag{186}$$

where $\phi_{TE,top}$ and $\phi_{TE,bottom}$ are the potential values at the trailing edges for the top and bottom points, respectively. In Figure 45, the values of the potential $\phi_{TE,top}$ and $\phi_{TE,bottom}$ are known, where the subscripts TE, top and $TE, bottom$ correspond to the trailing edge upper and lower i coordinates, respectively. Since these values are defined at the cell's center, and are not known at the trailing

edge, an extrapolation is needed. The simplest way to accomplish it is by using the Taylor series expansion. The value of ϕ at the trailing edge is computed from,

$$\begin{aligned} \phi_{TE,top} &= \phi_{face,top} + \triangle S \frac{\partial \phi_{face,top}}{\partial s}, \\ \phi_{TE,bottom} &= \phi_{face,bottom} + \triangle S \frac{\partial \phi_{face,bottom}}{\partial s} \end{aligned} . \tag{187}$$

The partial derivative is the tangential velocity at the trailing edge. The length $\triangle S$ is the distance from the face to the trailing edge. Practically, it is equal to half of the length of the face. Once the circulation value is calculated, the next step is to apply the potential jump. All the cells above and below the cut, which is drawn from the airfoil surface to the farfield boundary, will have the value of their potential modified to satisfy the proper jump condition. It is done as follows:

$$\begin{aligned} \phi_{i,0} &= \phi_{i,jmax} + \Gamma, \\ \phi_{i,-1} &= \phi_{i,jmax-1} + \Gamma, \\ \phi_{i,jmax+1} &= \phi_{i,1} - \Gamma, \\ \phi_{i,jmax+2} &= \phi_{i,2} - \Gamma. \end{aligned} \tag{188}$$

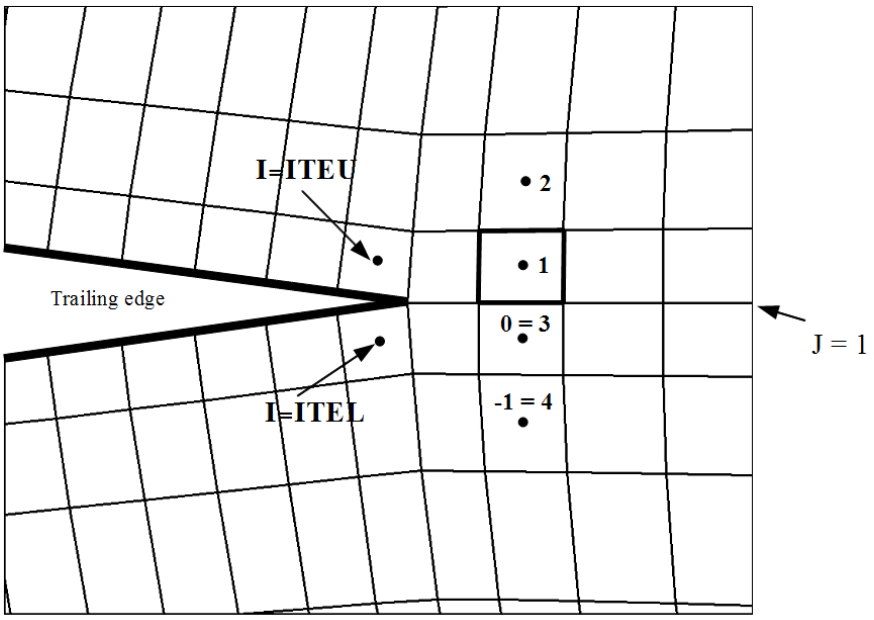


Figure 45: The Kutta-Joukowski condition in the finite volume method.

In this way, the flow over the airfoil must satisfy the Kutta condition, which says that the fluid flowing over the upper and lower surfaces of the airfoil meets

at the trailing edge of the airfoil. This condition explains how an inviscid fluid can generate lift. In reality, it is friction between the boundary of an airfoil and the fluid that allows for the flow to meet smoothly at the trailing edge. Thus, the Kutta condition accounts for friction at the boundary of an airfoil that is necessary for lift to be generated. The Kutta condition forces two additional constraints in the flow around an airfoil. First, the condition imposes that the leading and trailing edges of the airfoil are stagnation points [12]. Second, the angle of attack of the airfoil must remain well below a critical angle known as the stall angle. As the angle of attack increases beyond the stall angle, the Kutta condition is no longer physically applicable because the flow is no longer smooth and continuous.

It is important to mention that the Kutta-Joukowski theorem states a relationship between the circulation around an airfoil and the lifting force acting upon it. The relation is $\overrightarrow{L} = \rho \cdot \overrightarrow{V} \cdot \Gamma$ where L is the force (lift) experienced in the direction normal to the velocity V. This fact states that if there is no circulation around the airfoil we cannot have lift. We will not discuss this further and it is not part of this research work.

Remark: The flow around an airfoil was solved also in a C-grid approach but several obstacles were encountered, so that in the end the O-grid type was preferred. In order to describe how to implement the Kutta condition in the C-grid approach, let us look at some of the issues involved in more detail. Consider the case of the 2D airfoil with a sharp trailing edge surrounded by finite volumes, as sketched in Figure 46. Consider the five cells a, b, c, d, and e. For computing the divergence in cell a, for example, it is necessary to compute the fluxes on the edges of the cell that are not lying on the body. On this face, a solid wall boundary condition is applied, and this is done by $\partial\phi/\partial n = 0$. In practice, this condition enforces a ghost point value, which leads to a zero contravariant velocity $V_{\xi n}$. This cell face does not pose any special problem in implementing the solid wall boundary condition. Now consider the cell with centroid b. While imposing the solid wall boundary condition on the edge $1 - 2$, a problem arises because we do not have enough ghost points to impose the zero contravariant velocity. Since the velocity formula includes six points, from both sides of the edge, in this case , one of them is not a ghost point since it is located on the cut, where a periodic condition is applied. So in this case the solid wall boundary condition is not satisfied. Although this approach is slightly distorted, it worked fine in terms of convergence properties. Nevertheless, as mentioned before, an O-grid approach was preferred for this problem.

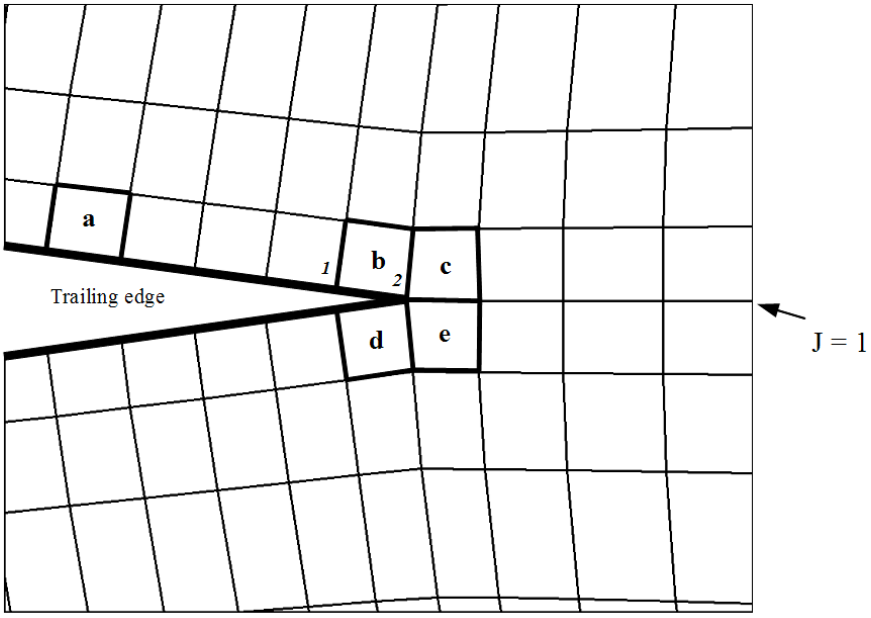


Figure 46: An implementation of the Kutta-Joukowski condition.

Three test cases are presented for the flow past an airfoil. These test cases can be categorized into two basic groups. First is the nonlifting case, NACA-0012 with a zero angle of attack $\alpha = 0°$, where a symmetric target pressure distribution is specified. The second group of test cases addresses the lifting airfoils, which include NACA-0012 with $\alpha = 1.25°$ and NACA-2822 with $\alpha = 0°$. The solution process of these two groups is slightly different.

12.6.4 Boundary conditions

For the airfoil surface, a solid wall boundary condition is imposed (no penetration),

$$\rho \overrightarrow{V} \cdot \hat{n} = \rho \frac{\partial \phi}{\partial n} = 0. \tag{189}$$

If the grid adjacent to the cut is orthogonal, the implementation of this condition is straightforward, and the following equation is satisfied:

$$\frac{\partial \phi}{\partial n} = \frac{\phi_{interior} - \phi_{ghost}}{\triangle \xi}, \tag{190}$$

where ϕ_{ghost} is the the potential value at the ghost cell, and $\triangle \xi$ is the distance between the interior cell center and the ghost cell's center. The condition is satisfied while $\phi_{interior} = \phi_{ghost}$. This case is relatively simple to implement. If the grid adjacent to the cut is not orthogonal, exactly as described in this flow problem, the solid wall boundary condition is applied as described in Section 12.

There are two ways in which the farfield boundary conditions are imposed. The first method, which was chosen to be implemented in this research work, is using a Neumann boundary condition at the farfield. The flow field is assumed to be known at the farfield and is taken as the free-stream condition. Practically, it is implemented by projecting the free-steam velocity V_∞ normal to the cell's face. A periodic boundary condition is applied to the cells located at the top and bottom of the cut. When lift is being produced (airfoil with an angle of attack or nonsymmetric airfoil), the Kutta condition is enforced by applying a jump in potential across the cut, from the trailing edge to the ghost cells at the farfield. An example of implementing the cut and farfield boundary conditions is given in Figure 47. This figure presents a partial view of the grid adjacent to the farfield boundary and the cut. The ghost cell's centers are sketched by the sign x. The ghost cells above and below the cut are formed as mirror reflections of the interior cells. In this way, it is ensured that the vector connecting the center of the ghost cells with the adjacent interior cell's center lies in the direction of the boundary normal vector. It is important for the Kutta condition to be satisfied. In order to compute the residual at point (i, j) we have to know first the potential values at the ghost cells. The potential value of cell (i, j) is set to the free-stream condition (Neumann condition). The Kutta condition is applied to all the cells above the cut and their values are modified to satisfy the proper jump condition, according to (188). This analysis holds true for the ghost cells in the opposite direction.

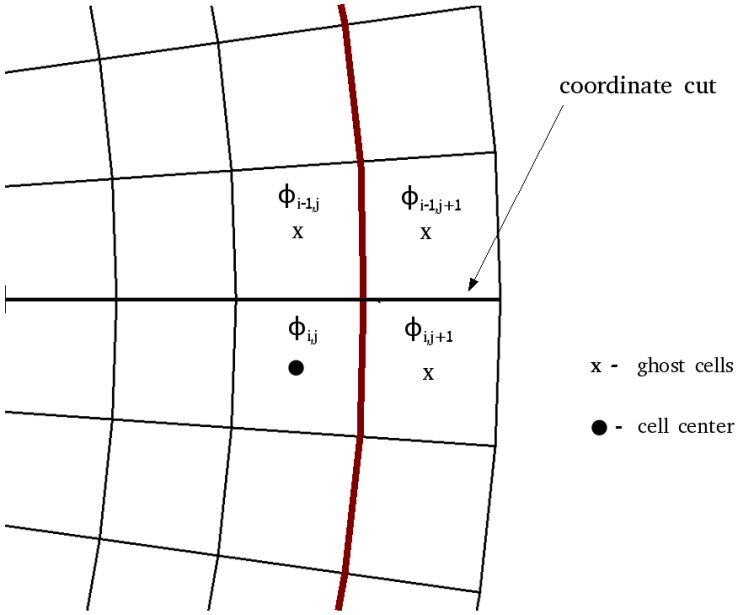


Figure 47: Illustration of how the Kutta condition is applied. A partial view of the airfoil O-type grid near the farfield boundary.

The second method to implement the Kutta condition is to apply the Dirichlet boundary condition at the farfield. This method is described by Jameson [26, 28,

31] and also by Hirsch [12]. Assuming a free-steam condition of an axial flow of V_∞ in the x-direction, the potential at the farfield can be expressed as

$$\phi = V_\infty \cdot x, \tag{191}$$

while the origin of the grid coordinate is taken as the airfoil trailing edge. When lift is being produced, the Kutta condition is enforced and then the above boundary condition is modified to take into account the circulation across the cut. The vortex of strength Γ is added to the boundary condition and it becomes,

$$\phi = V_\infty \cdot x + \frac{\Gamma\beta}{2\pi}. \tag{192}$$

The angle β is measured from the cut (which is aligned with the x-axis). Returning now to the first method – When applying the Neumann boundary condition, the additional term of the circulation is not reflected in the flux calculation across the boundary. This fact can be proved by rewriting Eq. (192) in a cylindrical coordinate system while $x = rcos\,(\beta)$,

$$\phi\,(r, \beta) = V_\infty \cdot rcos\,(\beta) + \frac{\Gamma\beta}{2\pi}. \tag{193}$$

Since the flux of a given face is computed in the normal direction, the change of the potential ϕ with respect to the radial direction (normal to the boundary faces) can be written as follows:

$$\frac{\partial\phi}{\partial r} = V_\infty cos\,(\beta)\,. \tag{194}$$

As one can see, when applying the Neumann boundary condition, the additional term of the circulation to the boundary condition is not reflected in the velocity value, normal to the cell's face. As mentioned, the first method was chosen to be implemented for cases of both NACA-0012 and NACA-2822. It proved to be efficient to code and was simple in its formulation.

12.6.5 NACA-0012 – qualitative results for $\alpha = 0°$

Before doing detailed quantitative analysis of the simulation results, the overall characteristics of flow field were examined to make sure that the solution reflected the expected behavior. All the problems were computed with 128×48 mesh size. Figure 48 demonstrates the results of test cases in which subsonic and transonic flows are involved. The Mach number around a NACA-0012 airfoil with a free-stream Mach number varies from 0.1 to 0.8 with a zero angle of attack, $\alpha = 0°$, as demonstrated. The first order upwind scheme has been applied in the supersonic flow regime. The results compare very well with Mach contour distributions in the literature [14, 12, 96], and also on grids with lower resolutions. The flow around the airfoil undergoes a strong local velocity gradient in the leading edge region, where over $2 - 3$ cells downstream of the stagnation point, the velocity increases from nearly zero to values larger than the incoming free-stream velocity. The following basic features are visible, indicating that qualitatively accurate results are being depicted:

- A stagnation point at the leading edge and trailing edge of the airfoil.

- Symmetric Mach contours at the top and bottom of the airfoil.

  The Mach 0.1 and 0.3 cases have only a slight difference in both the Mach contours and the C_p solution. The flow is close to being incompressible at these speeds, giving very little difference in the results. At $M_\infty = 0.5$ the compressibility of the flow become more significant and at $M_\infty = 0.76$ the compressibility effects are very evident.

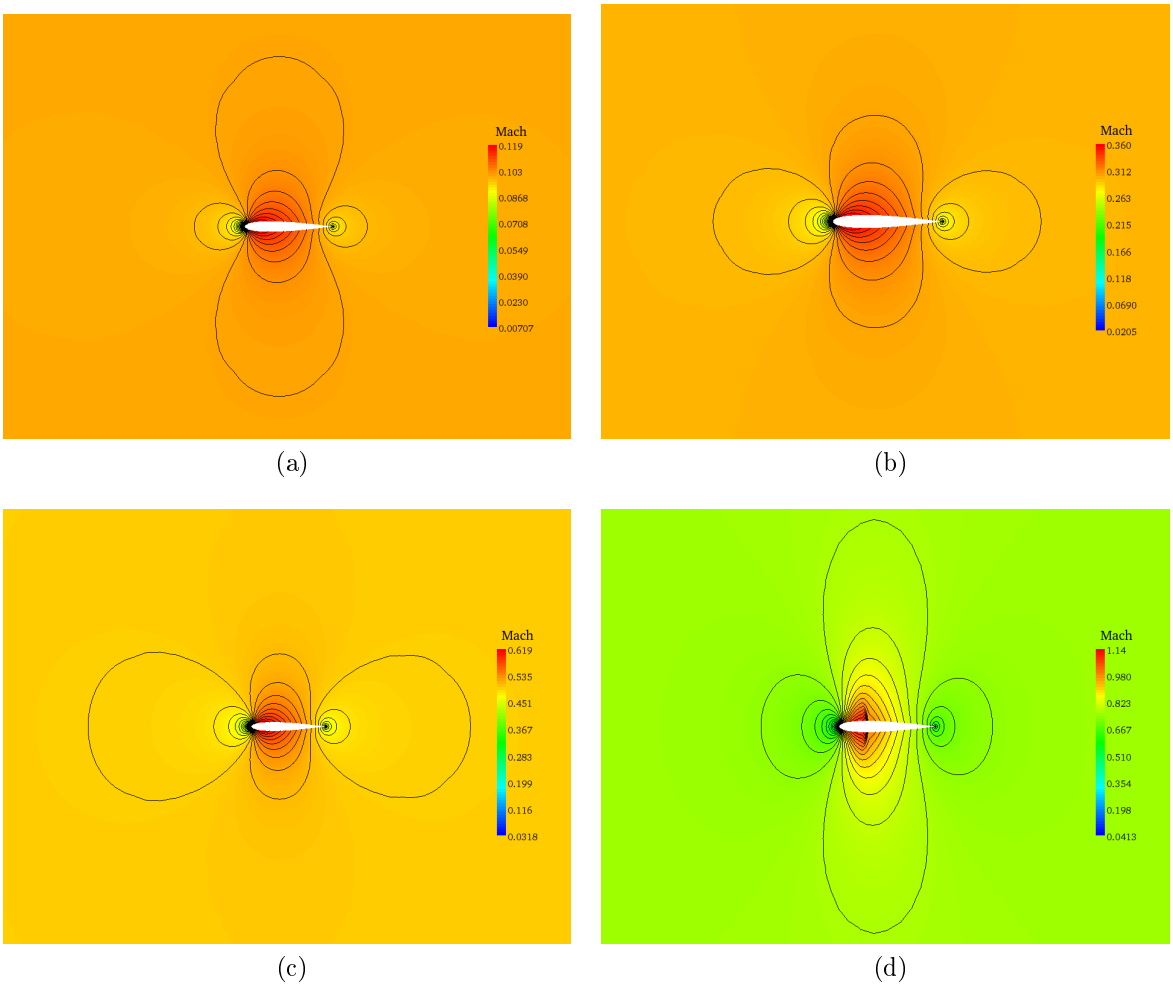


Figure 48: Distribution of the Mach contours as computed on the (128×48) mesh, for different incident Mach numbers. a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.76$. Observe the shock appearing at the top and bottom surfaces.

12.6.6 The pressure coefficient C_p

The aerodynamic performance of airfoil sections can be studied by reference to the distribution of pressure over the airfoil. This distribution is usually expressed in terms of the pressure coefficient:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{\frac{p}{p_\infty} - 1}{\frac{1}{2}\frac{\rho_\infty}{p_\infty}u_\infty^2} = \frac{\left(\frac{\rho}{\rho_\infty}\right)^\gamma - 1}{\frac{1}{2}\gamma M_\infty^2}. \tag{195}$$

The pressure coefficient C_p is the difference between the local static pressure and the free-stream static pressure, nondimensionalized by the free-stream dynamic pressure. The airfoil pressure distribution is generally plotted as C_p versus x, where the parameter x varies from 0 at the leading edge to 1 at the trailing edge. The pressure coefficient is plotted with negative values, higher on the top. This is done so that the upper surface of a conventional lifting airfoil corresponds to the upper curve. The C_p starts from about 1 at the stagnation point near the leading edge and rises rapidly (the pressure decreases) on both the upper and lower surfaces, and finally recovers to a small positive value of C_p near the trailing edge. The lift coefficient is related to the C_p by: $C_l = \int_0^x (C_{p_l} - C_{p_u})\, dx$, where C_{p_l} and C_{p_u} are lower and upper surface pressure, respectively. It is the area between the curves. Figure 49 shows the pressure distribution on the upper and lower surfaces computed by the pressure coefficient with negative C_p toward the top, following the usual convention. Observe the sharp decrease in C_p across the shock wave. The $M_\infty = 0.1$ and $M_\infty = 0.3$ cases have a slight difference in both the convergence and C_p solution. The flow is close to being incompressible at these speeds, giving very little difference in the results. At $M_\infty = 0.5$ the flow becomes more compressible and at $M_\infty = 0.76$ the compressibility effects are very evident.

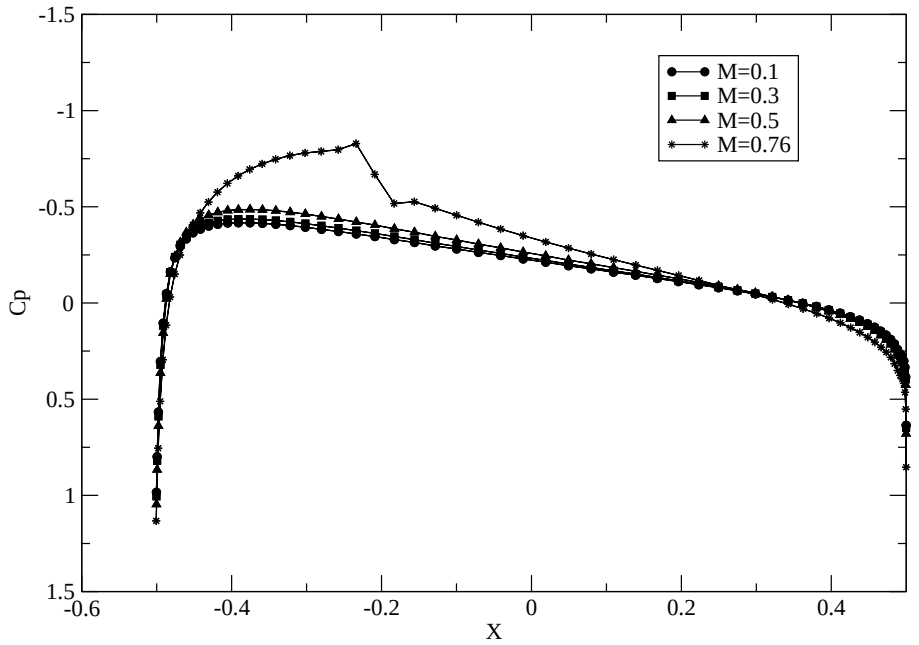


Figure 49: The distribution of pressure coefficient for an airfoil NACA-0012 as computed, using (128×28) mesh, for four Mach numbers.

12.6.7 AMG performance

In the subsonic cases, $M_\infty \leq 0.5$, the coarse-levels were obtained while applying a fixed threshold ($\varepsilon = 0.25$). It was sufficient to apply the second pass process in the first coarsening step and maintain standard coarsening for all subsequent levels. In the transonic case, $M_\infty = 0.76$, a dynamic threshold produced much better results (in terms of convergence properties) but at the expense of somewhat greater memory requirements.

The strength of connection between the matrix variables can be clearly seen from Figure 50, where the finest and first coarse-levels are plotted. For $M_\infty = 0.1$ the operator is nearly isotropic, and due to the setting of $\varepsilon = 0.25$, the AMG algorithm treats most of the connections contained in the corresponding matrix as strong (at least, there is an azimuthal symmetry in the strong connection). Consequently, the first coarse-level corresponds to red-black coarsening in most of the domains, far from the airfoil surface. There is a significant deviation from this coarsening pattern near the airfoil surface where an isotropic coarsening is observed. The reason is that the grid is highly stretched near the airfoil surface and the cell's aspect ratio becomes nearly unity. In a few cells above the airfoil the anisotropy is largest and this is well reflected in the one-dimensional coarsening pattern, while the AMG coarsened in the direction of the strong connectivity, in the η-direction. This coarsening pattern is essentially the same for the rest of the cases.

Remark: Also in this context, with a careful examination of the cells on the cut ($j = 1$, $j = jmax$), it can be seen that away from the cut or the airfoil surface the cells are roughly square, while this is not the case when approaching the cut or the airfoil surface. The grid is highly stretched. Looking at the discretization scheme, notice how the coefficients in the ξ-direction compare to the η-direction, especially above the cut. Their ratio scales as the grid cell aspect ratio. For example, if $S_\eta = 100 S_\xi$, then the η-direction coefficient would be 100 times larger than the ξ-direction coefficient. Consequently, a given change in the derivatives along the η-direction would influence the new solution 100 times more strongly than the same change in the derivative along ξ-direction. This has a very important effect that can be viewed from two aspects: The first one is that the error components in the large coefficient direction are reduced much more quickly than those in the small coefficient direction. So not only does SGS relaxation reduce fast error components, it does so significantly in the large coefficient's direction. Thus the slow error components in the small coefficient direction are reduced much more slowly. This directional sensitivity of the iterative solver is caused by the anisotropic coefficients, which can be a result of the equation or an irregular grid. The second aspect is that since the fast error components in large coefficients are reduced faster, it is well reflected in the coarsening process. The problem shows strong dependence in the η-direction, and little or no strong dependence in the ξ-direction. Perhaps the most important observation is that the grid has been coarsened only in the direction of strong dependence. Because smooth error varies slowly in the direction of strong dependence, interpolation can be performed accurately in that direction.

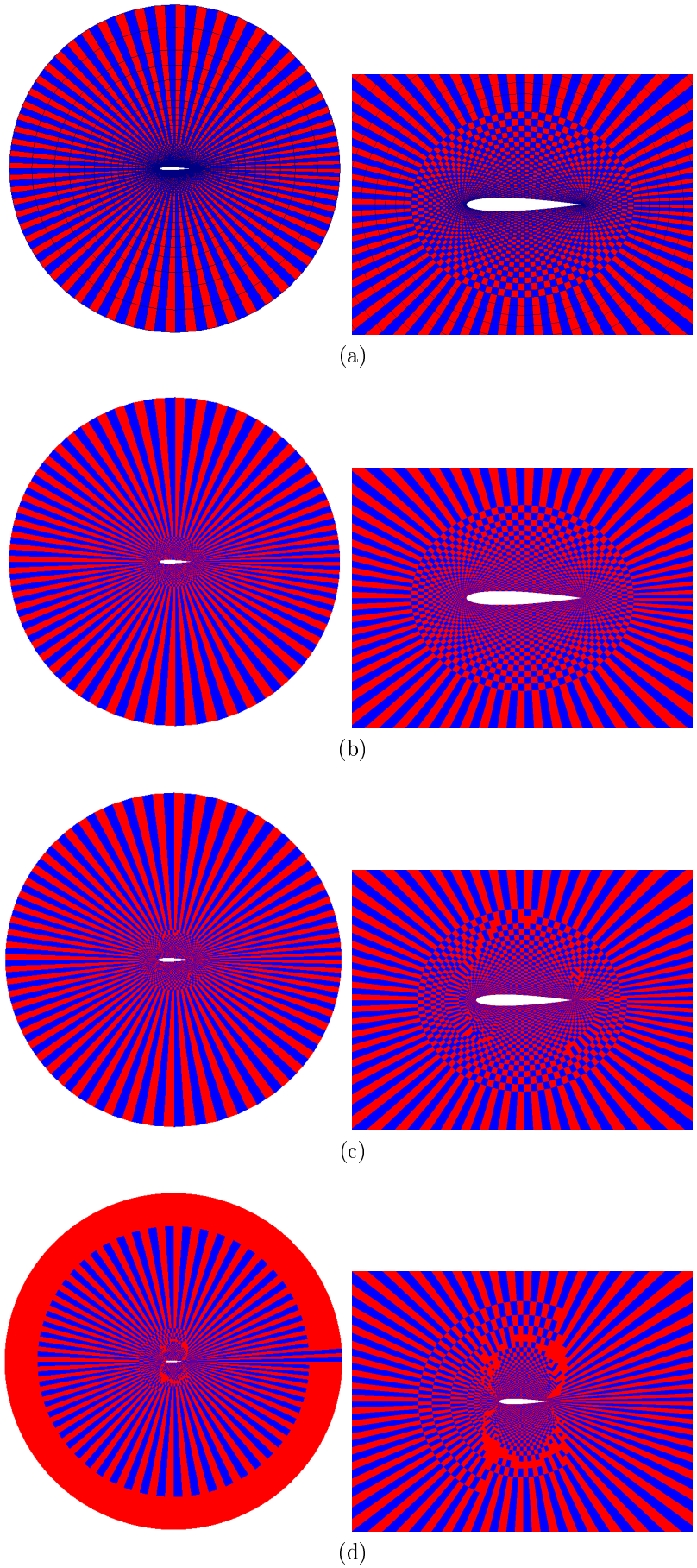


Figure 50: The finest and first coarse-level for mesh size (128×48). The red color corresponds to the C-point and the blue color corresponds to the F-point. The five flow conditions are as follows: a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.76$. The pictures on the right are magnified views of the airfoil region.

The discrete L_2-norm of the residual is shown in Table 20 for each V-cycle, in four different cases of Mach number. In Table 21 the grid and operator complexities are presented for the above Mach numbers. When the flow is subsonic, $M_\infty \leq 0.5$, the convergence rate is bounded below an order of magnitude. After 12 V-cycles a residual smaller than 10^{-10} was reached. In the subsonic cases, $M_\infty \leq 0.5$, the coefficient matrices A, $\tilde{A}$, and A^* were changed very slowly from one cycle to the next, especially above and below the airfoil's surface and at the leading and trailing edges, where the velocity gradient is maximal. Consequently, performing the setup phase only once, at the beginning of the solving phase, is sufficient to achieve a stable and efficient convergence; thus these cases were solved with 12 V-cycles.

The influence of compressibility on the overall convergence is well observed in the transonic case $M_\infty = 0.76$; where, as expected, convergence becomes considerably slower. The results were obtained by repeating the setup phase 6 times while applying two V-cycles between each update. This results in a total of 12 V-cycles until the residual decreased to the desired level of ~10^{-10}. Concerning the convergence behavior near the shock waves, it is generally observed that standard error smoothing is less efficient near the discontinuity. Therefore, there are two possibilities to improve convergence.

The first approach, which was introduced by Brandt [97], is to apply local smoothing sweeps in the neighborhood of the singularity. The additional work does not seriously affect the overall complexity and convergence properties since the number of points that forms the shock wave is usually very small in comparison with the number of interior points (set of measure zero). However, applying this approach results in a minor improvement in the overall convergence properties. The second approach that was chosen to be applied for this problem, is to invest more effort by additional smoothing sweeps (~1-2 local iterations) on the fine-level. This approach was preferred in order to improve the overall convergence properties in an inexpensive way and to overcome this convergence degradation. In addition, the main purpose of this "extra" work is mainly to "locate" the shock and to establish its profile, i.e. to "clean-up" the area of the strong nonlinearity as the coarse-level correction is distributed on the fine-level. In this way we also significantly reduce the accumulation of errors that were interpolated from the coarse-levels.

The convergence histories are demonstrated in Figure 51. It is clear that as the Mach number increases so is the number of iterations required to solve the problem. We observe that for subsonic cases, $M_\infty = 0.1 - 0.5$, we need less than 10 V-cycles to reduce the L_2-norm of the residual by 10 orders of magnitude. As for the transonic case of $M_\infty = 0.76$, the convergence is somewhat slower here while about 12 cycles are required to reduce the residual by 10 orders of magnitude.

This extra increase in the number of iterations is mainly due to the strong anisotropy and the existence of two supersonic pockets that are terminated by shock waves. For $M_\infty = 0.76$, the problem was converged less than an order of magnitude per V-cycle, and a much greater operator complexity of $C_\Omega = 3.89$ was obtained. This behavior is typical of the more general problems, which are characterized by a strong anisotropy and nonlinearity of the matrix coefficients. It is important to mention that according to our particular requirements, it is too

high but still practical. It is not surprising as shown in Table 20 and Table 21, that you get what you pay for. Combinations of AMG components that produce the best convergence rates are also those with the greater costs. Parameter selection is largely the art of finding a compromise between performance and cost.

Table 20: The results of AMG V-cycles applied to the flow through a symmetrical NACA-0012 airfoil. The second norm of the residual after each V-cycle is presented for 128×28 mesh size.

	$M_\infty = 0.1$		$M_\infty = 0.3$		$M_\infty = 0.5$		$M_\infty = 0.76$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	7.095e-04	–	2.276e-03	–	5.015e-02	–	8.959e-03	–
1	2.956e-05	0.04	1.631e-04	0.07	9.031e-04	0.02	1.722e-03	0.19
2	7.825e-07	0.03	9.298e-06	0.06	6.360e-05	0.07	7.216e-04	0.42
3	2.611e-08	0.03	4.834e-07	0.05	6.443e-06	0.10	4.419e-04	0.61
4	9.265e-10	0.04	2.726e-08	0.06	6.322e-07	0.10	1.017e-04	0.23
5	4.162e-11	0.04	1.421e-09	0.05	7.259e-08	0.11	1.853e-05	0.18
6	2.314e-12	0.06	7.824e-11	0.06	8.328e-09	0.11	4.673e-06	0.25
7	3.898e-13	0.17	4.293e-12	0.05	1.008e-09	0.12	1.027e-06	0.22
8	3.627e-13	0.93	6.056e-13	0.14	1.217e-10	0.12	1.993e-07	0.19
9	–	–	5.464e-13	0.90	1.487e-11	0.12	3.810e-08	0.19
10	–	–	–	–	1.809e-12	0.12	7.416e-09	0.19
11	–	–	–	–	2.210e-13	0.12	1.456e-09	0.20
12	–	–	–	–	3.289e-14	0.15	2.857e-10	0.20

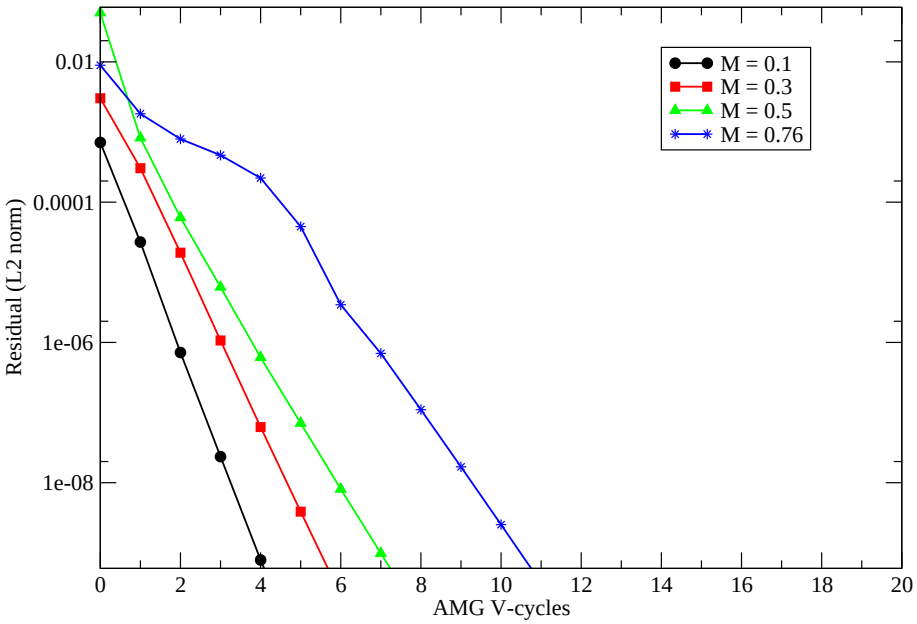


Figure 51: Convergence histories.

Table 21: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

Complexities	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.5$	$M_\infty = 0.76$
C_Ω	1.90	1.93	2.13	1.92
C_L	3.01	2.95	3.87	3.89

This problem was solved for two more grid sizes of 64×24 and 32×12. Convergence factor C_f and L_2-norm of the residual $\|R^m\|_2$ are presented in Table 22. This is a case where applying AMG convergence acceleration is a very attractive possibility since it is precisely a situation, for which constructing an efficient geometric multigrid approach would become extremely cumbersome. We observe that for subsonic flow the AMG exhibits the same type of convergence that was observed in the previous problems. The residual L_2-norm is decreased by a relatively constant factor, below an order of magnitude per V-cycle. It takes about 8 V-cycles to reach a residual of 10^{-12}. As the Mach number increases the convergence deteriorates, but we find the solver to still be very efficient for this problem, despite of the strong nonlinearity of the problem. The storage requirement is greater than that for the subsonic case (isotropic case), the reason being that AMG performs one-dimensional coarsening in the azimuthal direction at the farfield since the

strong connection in the ξ-direction (azimuthal direction) arises from a large grid spacing in η- direction. In this respect, the complexities deteriorate significantly.

Table 22: The results of AMG V-cycles applied to the flow through an NACA-0012 airfoil. The second norm of the residual after each V-cycle $\|R^m\|_2$ and the convergence factor C_f are detailed for two different mesh sizes.

	$M_\infty = 0.1$				$M_\infty = 0.76$			
grid size $\longrightarrow$	33×12		65×25		33×12		65×25	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	1.368e-03	–	1.347e-03	–	6.849e-03	–	1.123e-02	–
1	6.872e-05	0.05	5.310e-05	0.39	1.104e-03	0.16	2.447e-03	0.21
2	3.208e-06	0.05	1.956e-06	0.04	1.720e-04	0.16	9.265e-04	0.38
3	1.461e-07	0.05	1.271e-07	0.06	1.862e-05	0.11	3.680e-04	0.40
4	6.702e-09	0.05	8.625e-09	0.07	3.282e-06	0.18	1.355e-04	0.37
5	3.273e-10	0.05	5.882e-10	0.07	6.709e-07	0.20	4.198e-06	0.03
6	1.777e-11	0.05	4.058e-11	0.07	9.692e-08	0.14	1.885e-07	0.04
7	1.138e-12	0.06	2.843e-12	0.07	1.079e-08	0.11	8.792e-09	0.05
8	3.787e-13	0.33	2.042e-13	0.07	9.245e-10	0.09	5.410e-10	0.06
9	3.717e-13	0.98	3.190e-14	0.16	5.690e-11	0.06	3.702e-11	0.07
10	3.716e-13	1.00	2.837e-14	0.89	5.028e-12	0.09	2.695e-12	0.07
11	–	–	–	–	9.383e-13	0.19	7.177e-14	0.03
12	–	–	–	–	1.226e-13	0.13	2.322e-14	0.32
C_Ω	**1.88**		**1.91**		**2.00**		**2.09**	
C_L	**2.62**		**2.93**		**3.18**		**3.54**	

Let us examine the storage requirements for two cases of $M_\infty = 0.76$ and $M_\infty = 0.1$. For $M_\infty = 0.1$. The problem was solved using six multigrid levels and the solver's characteristics are presented in Table 23. The initial coarse-level, A^{m+1}, has 3072 points, exactly half the number of points on the fine-level. This can be clearly seen in the coarsening pattern presented in Figure 50(a). The coarsening in the next coarse-levels is more "aggressive", while each succeeding coarse-level has fewer than half the number of points as the next finer level. It can be seen in the last column that the average number of nonzero entries per row increases as the grids become coarser. For $M_\infty = 0.76$, the situation is even more dramatic. In the last column it can be seen that the average number of nonzeros per row increases as the grids become coarser. However, the operator and grid complexities are slightly damaged.

Table 23: Results of AMG V-cycles applied to the flow around NACA-0012 airfoil.

	Number of rows		Number of non-zeros		Average entries per row	
Level	$M_\infty = 0.1$	$M_\infty = 0.76$	$M_\infty = 0.1$	$M_\infty = 0.76$	$M_\infty = 0.1$	$M_\infty = 0.76$
A^m	6144	6144	108983	110289	17.7	17.9
A^{m+1}	3072	3323	104678	148812	34	44.7
A^{m+2}	1437	1758	60275	86978	41.9	49.4
A^{m+3}	659	968	33783	45543	51.26	47
A^{m+4}	266	532	14869	23786	55.89	44.7
A^{m+5}	104	278	5805	12115	55.8	43.5

We would like to recall here that our main goal in these applications is to demonstrate how different AMG components can influence the overall performance. We have not tried to find a combination that results in an optimal performance, but rather confine ourselves to a set of components or parameters that result in an acceptable performance and in robust multi-purpose algorithm. The above results clearly show that an optimized parameter setting (for instance: dynamic threshold, matrix reduction parameter η, second-pass process), depending on the application, may improve the performance substantially further.

12.6.8 NACA 0012 - results for $\alpha = 1.25°$

Before doing detailed quantitative analysis of the simulation results, the overall characteristics of flow field were examined to make sure that the solution reflected the expected behavior. The following results were computed on the 128×48 mesh. Figure 52 presents the results of test cases in which subsonic and transonic flows are involved. The Mach number of isolines, for free-stream Mach number, varies from 0.1 to 0.71 with $\alpha = 1.25°$ demonstrated. It is well observed that the Kutta condition is satisfied and the stagnation point is located on the trailing edge. Figure 53 shows the pressure coefficient C_p on the upper and lower surfaces with negative values toward the top, following the usual convention. The presence of shock waves above and below the airfoil is well observed in the sharp decrease of C_p when $M_\infty = 0.71$.

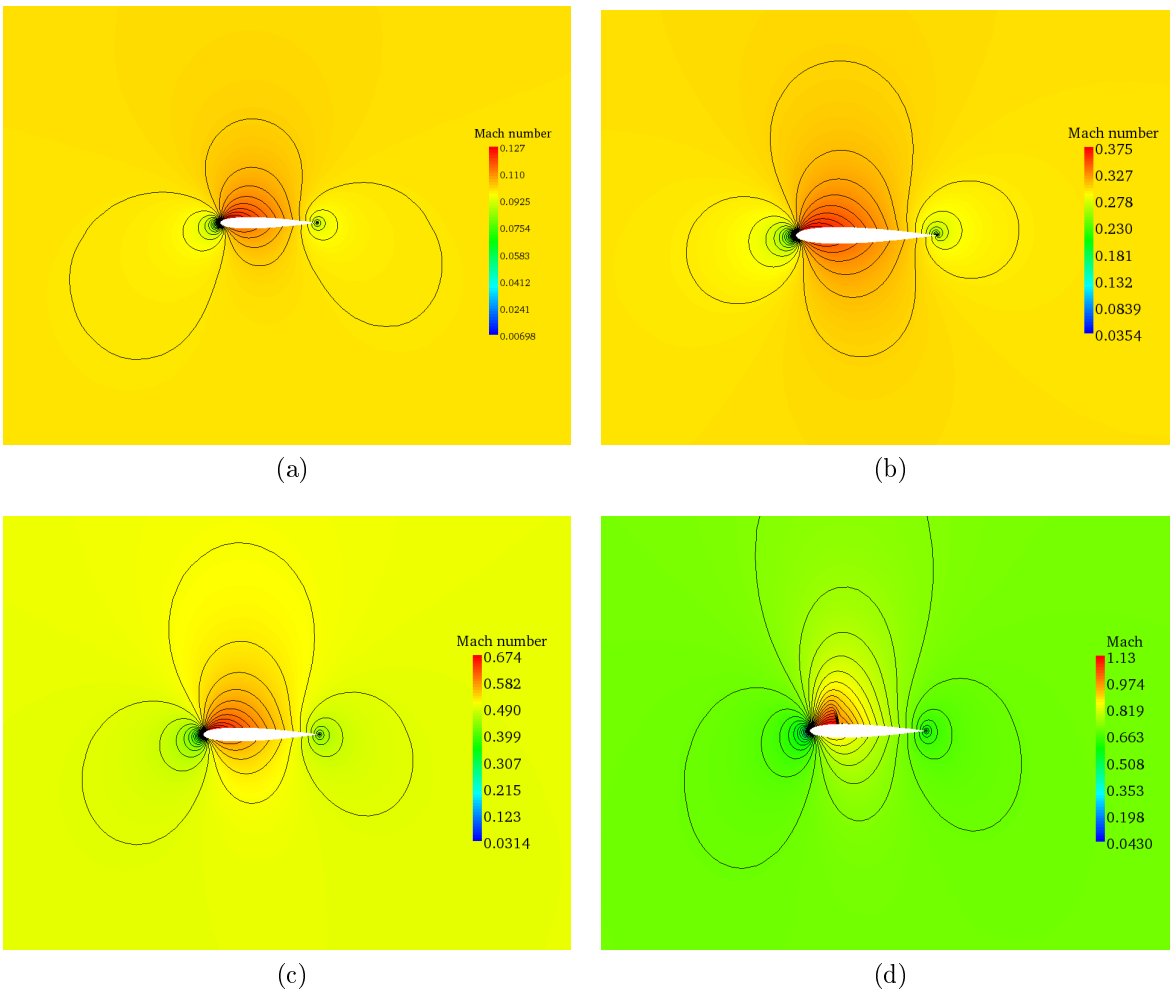


Figure 52: Distribution of velocity as computed on the (128×48) mesh, for different free-stream Mach numbers. a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.71$. Observe the shock appearing at top and bottom surfaces.

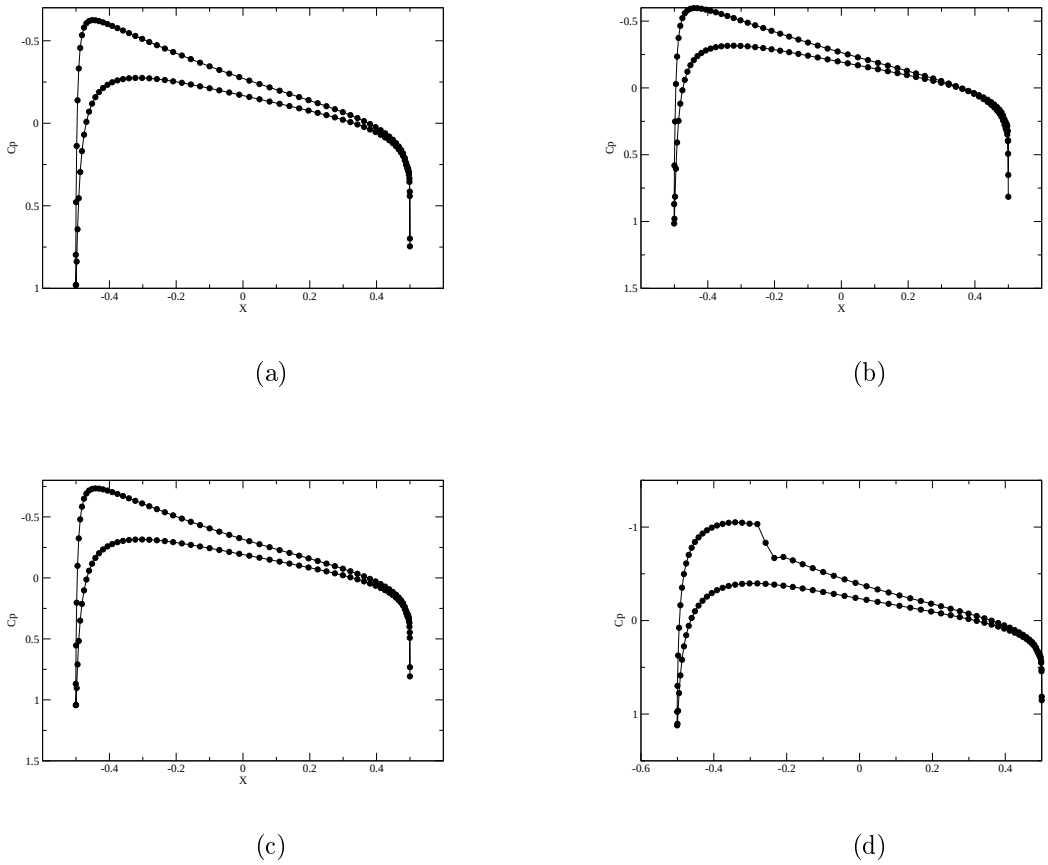


Figure 53: The distribution of pressure coefficient C_p for an NACA-0012 airfoil at $\alpha = 1.25°$as computed using (128×28) mesh, for four Mach numbers.

Table 24: Results of the circulation Γ for four cases of Mach number.

Circulation	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.5$	$M_\infty = 0.71$
Γ	7.83e-03	2.48e-02	4.73e-02	9.36e-02

12.6.9 AMG performance

A symmetrical NACA-0012 airfoil with $\alpha = 1.25°$ produces lift due to the imposed finite circulation Γ around the airfoil. The value of the circulation is unique for each set of flow condition. This problem is much more complicated to solve with the AMG method, because the circulation value must be updated after each smoothing sweep in order to satisfy the Kutta condition at the airfoil trailing edge. In fact, this problem becomes a kind of an optimization problem while the goal is to improve the circulation until convergence is reached for both the circulation Γ and the equation itself (the residual has to reach the level of discretization). Two

very important observations are in order here. First, since Γ is updated only on the fine-level, the overall AMG performance suffers significantly. The reason is that in order to solve the problem, both the residual and the circulation have to reach convergence. Second, however, each circulation update results in a new set of potential values along the cut, which leads to a somewhat different solution and, therefore, has an extreme impact on the overall convergence of the algorithm. The coarse-level correction loses its relevance at the moment the circulation is updated. As a consequence, the convergence rate becomes extremely slow (around 0.85). This action of changing the boundary values (along the cut) is against the basic ground rules of the multigrid method. This global process gives rise to slow error components, which results in unsatisfactory AMG behavior. Note that in the elliptic problem (subsonic flow), when imposing a potential disturbance inside the domain or along the boundaries, all the points in the domain are influenced and "feel" the potential disturbance. Therefore, it is expected that this process could be extremely amplified when the potential disturbance occurs along the entire cut, which is constructed of several cells. It is important to note here that it does not matter if the residual is dropped five orders of magnitude (or even more) by successive V-cycles (without updating Γ). Once the circulation value is modified, the residual jumps three or four orders of magnitude and most of what we have gained in the AMG V-cycles is lost! Note that as the circulation Γ is "frozen" (at any Mach number mentioned above), the AMG performance becomes visible again with the convergence rate below an order of magnitude per V-cycle. However, this situation makes use of the AMG algorithm very unpractical.

Practical experience has shown that in order to "cure" this problem a possible way is to obtain a good approximation of Γ as an initial condition for the algorithm. The best way to obtain a good approximation Γ_0 is by solving the problem on a coarser grid (usually a geometrically four-times coarser grid is enough to obtain very fast convergence). This is done by the FMG method in the context of AMG, as detailed in Section 8. However, a properly designed FMG scheme can be much more effective in general than just sequence of V-cycles. Recall that the alternative of convergence rates around 0.9 are definitely not acceptable. The key principle regarding the FMG approach in general approach is that before the Ω^m problem is even touched, the Ω^{2m} problem has been solved to the level of discretization error and the result is a good approximation to the circulation value Γ_0 (its value is within $\sim 1-3\%$ of the value of Γ that obtained on the fine-level Ω^m) as an initial condition. It is very important to mention that the convergence rate of these nonlinear iterations (especially in the transonic case where the nonlinearity is significant) depends dramatically on a quality of the initial condition for Γ. The better the initial guess, Γ_0, used on the fine-level, the more effective the AMG V-cycle will be. In our test-case the FMG scheme relies on, four levels, while on the coarsest level (16×6 grid) the problem was solved to the level of machine zero.

Therefore, the practical implementation of this problem, relies heavily upon the way the treatment of the circulation value is incorporated in each V-cycle. The value of the circulation Γ must be updated at the end of each V-cycle, immediately after the correction from the coarse-levels is distributed to the points on the fine-level. In order to improve the speed of convergence, a robust choice of the circulation optimization must be made. Three possibilities are in order: In the first, which is the one chosen in this work, the circulation can be improved

by a simple 2D linear extrapolation of the circulation value. If the circulation value is known in the present and last iterations, the next value can be calculated as follows:, an over-relaxation parameter $1 \leq \omega \leq 2$ can be added in order to accelerate convergence:

$$\Gamma_{new} = \Gamma_{n+1} + \omega \left(\Gamma_{n+1} - \Gamma_n \right). \tag{196}$$

The second possibility is to apply an over-relaxation on Γ by the following formula,

$$\Gamma_{new} = \omega \Gamma_{n+1} + (1 - \omega) \Gamma_n, \tag{197}$$

where also in this case $1 \leq \omega \leq 2$. The third possibility is to first solve the problem on a coarse-level to the level of discretization and obtain the circulation value Γ_0. Then apply an under-relaxation ($\omega = 0.5$) on Γ by the following formula,

$$\Gamma_{new} = \omega \Gamma_0 + (1 - \omega) \Gamma_{n+1}. \tag{198}$$

The three methods were tested while the third Eq. (198), gave much better results in terms of convergence properties. However, it requires a pre-calculation of Γ_0 on a coarse-level.

Another technique that significantly improves the convergence is by freezing the value of Γ once it ceases to change by a certain amount. This is accomplished by monitoring each new update of Γ and calculating the percent change of the new value from previous one.

$$\Gamma_{change} = \frac{\Gamma_{n+1} - \Gamma_n}{\Gamma_{n+1}} \times 100. \tag{199}$$

When $\Gamma_{change} \approx 10^{-4}$, the value of Γ is frozen for the remainder of the AMG V-cycles.

In practice, the improved FMG is very effective for this specific problem because a very good initial guess is obtained on the fine-level Ω^m, although it costs a little more per cycle than the V-cycle scheme. The first coarse-level for each case described above is sketched in Figure 54 as a magnified view of the area around the airfoil. When the flow is incompressible, the operator is nearly isotropic and the points that construct the coarse-level are distributed uniformly, as can be seen in Figure 54(a). As M_∞ is increased the equation becomes more anisotropic and it is well reflected in the coarsening pattern where the number of coarse points is increased around the sonic line, above and below the airfoil, where the anisotropy is largest.

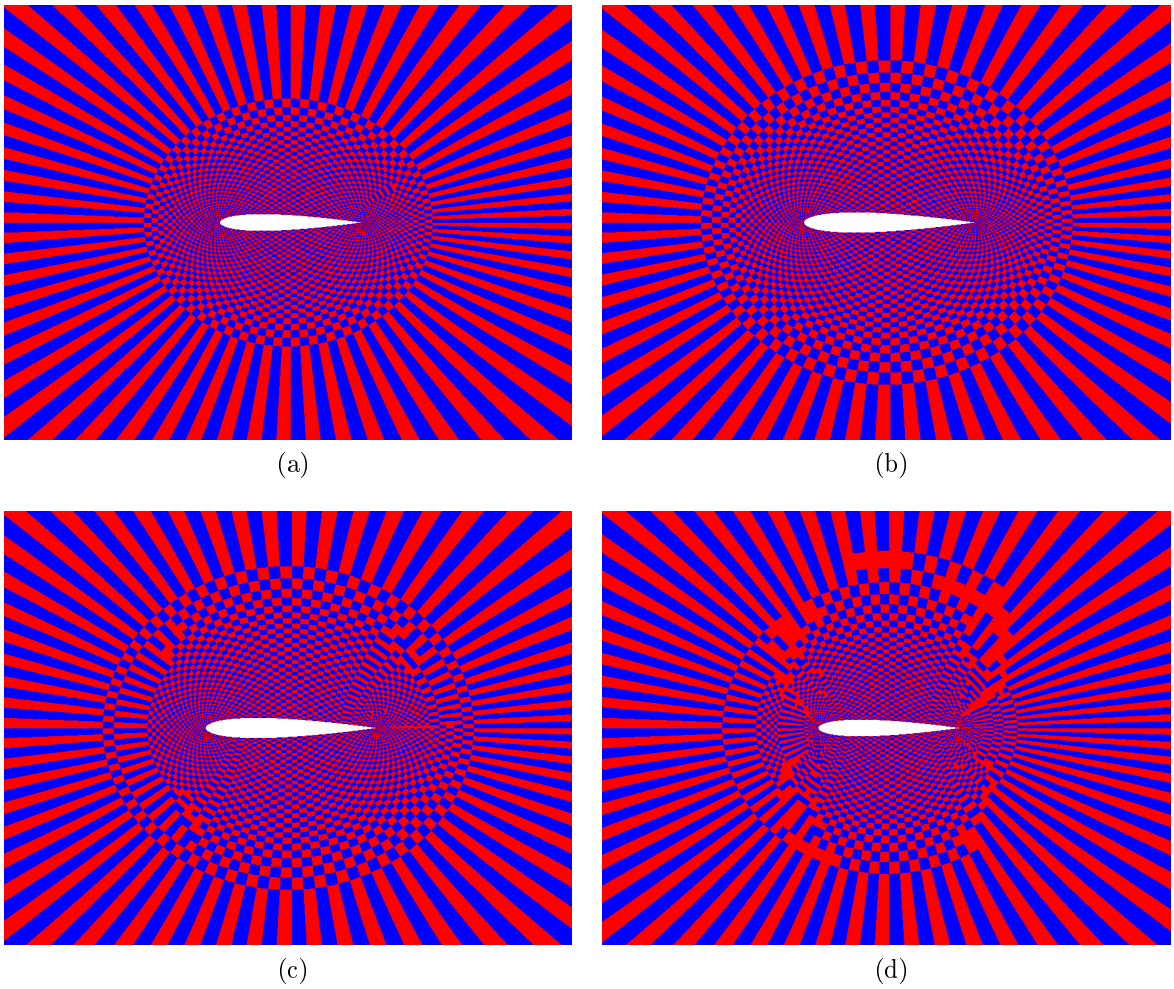


Figure 54: The fine and first coarse-level of the airfoil with $\alpha = 1.25°$ for four cases of Mach number: a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.71$. Blue point indicates an F-point and red point indicates a C-point.

The discrete L_2-norm of the residual and the convergence rate are shown in Table 25 for each V-cycle, in four different cases of Mach numbers. In Table 26 the grid and operator complexities are presented for the above Mach numbers. When the flow is subsonic, $M_\infty \leq 0.5$, the convergence rate is bounded below an order of magnitude. After 12 V-cycles a residual smaller than 10^{-10} was reached. In the subsonic cases, $M_\infty \leq 0.5$, the coefficient matrices A, $\tilde{A}$, and A^* were changed very slowly from one cycle to the next, even, above and below the airfoil surface and at the leading and trailing edges where the velocity gradient is maximal. Consequently, performing the setup phase only once, at the beginning of the solving phase, is sufficient to obtain a stable algorithm and efficient convergence. These problems were solved to the machine zero by applying 12 V-cycles on the finest level.

The influence of the compressibility on the overall performance is well observed in the transonic case $M_\infty = 0.76$; The convergence, as expected, becomes considerably slower. The results were obtained by repeating the setup phase six times while applying two V-cycles between each update. This results in a total of 12

V-cycles until the residual decreased to the desired level of ~10^{-10}. In this particular case, more effort was invested by additional local smoothing sweeps (~2 local iterations) in the neighborhood of the shocks. This appeared very helpful in order to improve the overall performance.

The convergence histories for the five cases are depicted in Figure 55. The effect of the Mach number on the AMG performance is clearly visible. The case of $M_\infty = 0.76$ is characterized by two supersonic pockets, which are terminated by shock waves above and below the airfoil. Convergence is somewhat slower here. It requires nearly 11 V-cycles to decrease the L_2-norm of the residual to a level of 10^{-10}. In the case of $M_\infty = 0.01$, they converged just by $7 - 8$ cycles. Although the problem was converged rapidly, this advantage is eaten up by the much greater complexities of $C_\Omega = 2.05$ and $C_L = 4.72$. This fact is typical for the more general problems, which are characterized with a strong anisotropy and nonlinearity of the matrix coefficients.

Table 25: The results of AMG V-cycles applied to the flow through a symmetrical NACA-0012 airfoil. The second norm of the residual $\|R^m\|_2$ after each V-cycle and the convergence rate C_f are presented for 128×28 mesh size.

	$M_\infty = 0.1$		$M_\infty = 0.3$		$M_\infty = 0.5$		$M_\infty = 0.71$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	2.090e-02	–	6.272e-02	–	0.104e-00	–	8.944e-02	–
1	1.824e-03	0.08	2.020e-02	0.32	2.708e-02	0.26	1.386e-02	0.15
2	1.873e-04	0.10	1.715e-03	0.08	2.912e-03	0.11	2.209e-03	0.16
3	1.573e-05	0.08	1.478e-04	0.09	2.471e-04	0.08	3.438e-04	0.16
4	1.281e-06	0.08	1.313e-05	0.09	2.270e-05	0.09	6.087e-05	0.16
5	1.018e-07	0.08	1.145e-06	0.09	2.199e-06	0.10	1.101e-05	0.18
6	8.159e-09	0.08	1.001e-07	0.09	2.294e-07	0.10	1.988e-06	0.18
7	6.560e-10	0.08	8.804e-09	0.09	2.100e-08	0.09	3.593e-07	0.18
8	5.318e-11	0.08	7.740e-10	0.09	1.756e-09	0.08	6.499e-08	0.18
9	4.332e-12	0.08	6.782e-11	0.09	1.745e-10	0.10	1.179e-08	0.18
10	3.508e-13	0.08	5.852e-12	0.09	1.968e-11	0.11	2.149e-09	0.18
11	3.956e-14	0.11	4.221e-13	0.07	1.614e-12	0.08	3.930e-10	0.18
12	3.170e-14	0.80	1.231e-13	0.29	1.152e-13	0.07	7.084e-11	0.18

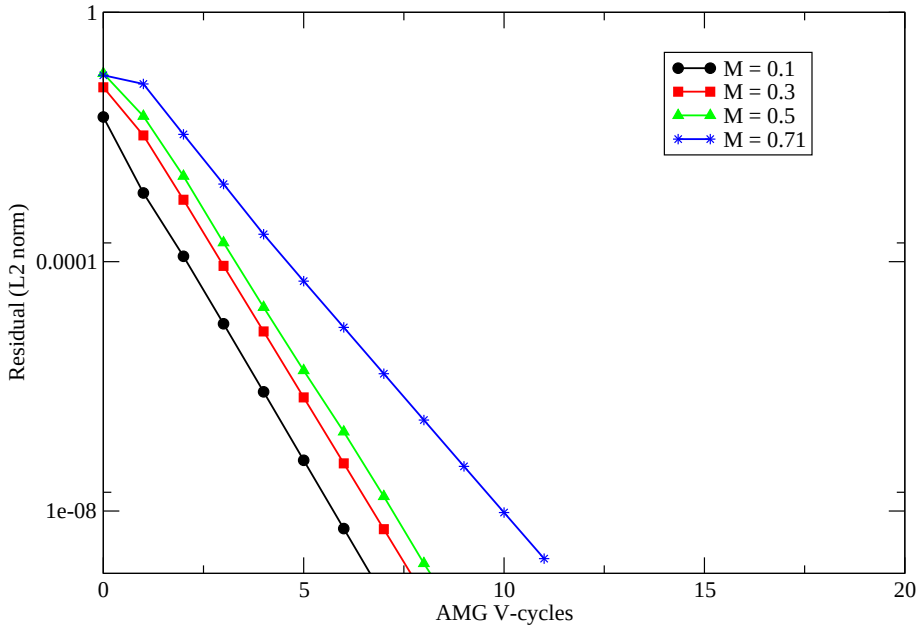


Figure 55: Convergence histories.

Table 26: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

Complexities	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.5$	$M_\infty = 0.71$
C_Ω	3.02	2.94	1.95	2.05
C_L	1.90	1.92	2.99	4.72

This problem was solved for two more grid sizes of 64×24 and 32×12. Convergence factor C_f and L_2-norm of the residual $\|R^m\|_2$ are presented in Table 27. For $M_\infty < 0.5$ we observe that for subsonic flow the AMG exhibits the same type of convergence that was observed for this problem solved on the finer grid. The residual L_2-norm decreases by a nearly constant factor, below an order of magnitude per V-cycle. It takes about 8 V-cycles to reach a residual of 10^{-12}. As the Mach number is increased the convergence degrades, but we find it still very good for this problem, inspite of the strong anisotropy. The storage requirements are greater than that for the subsonic case (isotropic case), the reason being that AMG performs one-dimensional coarsening in the azimuth direction at the farfield. This is because the strong connections in the ξ-direction (azimuth direction) arises from a large grid spacing in the η-direction. This fact strongly has a rather severe adverse effect on the complexities.

Table 27: The results of AMG V-cycles applied to the flow around NACA-0012 airfoil with $\alpha = 1.25°$. The second norm of the residual $\|R^m\|_2$ after each V-cycle and the convergence factor C_f are detailed for two different mesh sizes, 32×12 and 64×24.

	$M_\infty = 0.1$				$M_\infty = 0.3$			
grid size $\longrightarrow$	32×12		64×24		32×12		64×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	2.623e-02	–	2.324e-02	–	7.697e-02	–	6.900e-02	–
1	2.243e-03	0.08	2.324e-03	0.10	8.830e-03	0.11	1.188e-02	0.17
2	1.756e-04	0.08	2.259e-04	0.10	9.205e-04	0.10	1.106e-03	0.09
3	1.264e-05	0.07	1.916e-05	0.08	7.204e-05	0.08	8.374e-05	0.08
4	8.909e-07	0.07	1.544e-06	0.08	5.429e-06	0.08	6.317e-06	0.08
5	6.277e-08	0.07	1.229e-07	0.08	4.072e-07	0.08	4.776e-07	0.08
6	4.440e-09	0.07	9.787e-09	0.08	3.059e-08	0.08	3.613e-08	0.08
7	3.149e-10	0.07	7.811e-10	0.08	2.304e-09	0.08	2.735e-09	0.08
8	2.234e-11	0.07	6.252e-11	0.08	1.739e-10	0.08	2.074e-10	0.08
9	1.583e-12	0.07	5.016e-12	0.08	1.315e-11	0.08	1.574e-11	0.08
10	1.117e-13	0.07	4.016e-13	0.08	9.931e-13	0.08	1.175e-12	0.07
11	–	–	3.125e-14	0.08	–	–	7.187e-14	0.06
12	–	–	5.638e-15	0.18	–	–	3.233e-14	0.45
C_Ω	**1.85**		**1.91**		**1.88**		**1.91**	
C_L	**2.53**		**2.89**		**2.51**		**2.78**	

	$M_\infty = 0.5$				$M_\infty = 0.71$			
grid size $\longrightarrow$	32×12		64×24		32×12		64×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	0.135e-00	–	0.132e-00	–	7.285e-03	–	1.241e-02	–
1	2.340e-02	0.17	6.005e-02	0.04	4.031e-03	0.55	1.027e-02	0.82
2	2.104e-03	0.09	7.292e-03	0.12	1.707e-04	0.04	1.637e-03	0.15
3	1.557e-04	0.07	7.900e-04	0.11	9.955e-06	0.06	2.408e-04	0.15
4	1.180e-05	0.08	8.441e-05	0.11	1.062e-06	0.11	3.504e-05	0.15
5	8.979e-07	0.08	9.001e-06	0.11	1.822e-07	0.17	5.247e-06	0.15
6	6.779e-08	0.08	9.653e-07	0.11	1.996e-08	0.11	8.001e-07	0.16
7	5.082e-09	0.07	1.036e-07	0.11	3.331e-09	0.17	1.246e-07	0.16
8	3.788e-10	0.07	1.109e-08	0.11	3.970e-10	0.12	1.946e-08	0.16
9	2.808e-11	0.07	1.186e-09	0.11	4.774e-11	0.12	3.046e-09	0.16
10	2.053e-12	0.07	1.264e-10	0.11	1.522e-11	0.32	4.772e-10	0.16
11	–	–	1.331e-11	0.11	1.817e-11	1.1	7.469e-11	0.16
12	–	–	1.255e-12	0.09	–	–	1.161e-11	0.16
C_Ω	**1.95**		**1.92**		**1.99**		**1.99**	
C_L	**2.70**		**2.80**		**3.78**		**4.32**	

Table 28: The table shows the FMG performance applied to the flow around a NACA-0012 airfoil with $\alpha = 1.25°$ for four Mach numbers. The FMG(1,1) scheme is shown for one SGS relaxation on the descent phase and one SGS relaxation on the ascent phase. The FMG(1.1) number of levels is presented in the first row. The second row includes the AMG V-cycles performed in the FMG process. The average convergence factor between successive V-cycles is presented in the third row.

	$M_\infty = 0.1$			$M_\infty = 0.3$		
grid size $\longrightarrow$	32×12	64×24	128×48	32×12	64×24	128×48
FMG(1,1), number of levels	3	3	4	3	3	4
AMG V(1,1) - cycles	1	1	5	1	1	5
Average convergence factor,C_f	–	–	0.12	–	–	0.12
	$M_\infty = 0.5$			$M_\infty = 0.71$		
FMG(1,1), number of levels	3	3	4	3	3	4
AMG V(1,1) - cycles	1	1	5	1	10	10
Average convergence factor,C_f	–	–	0.13	–	0.12	0.14

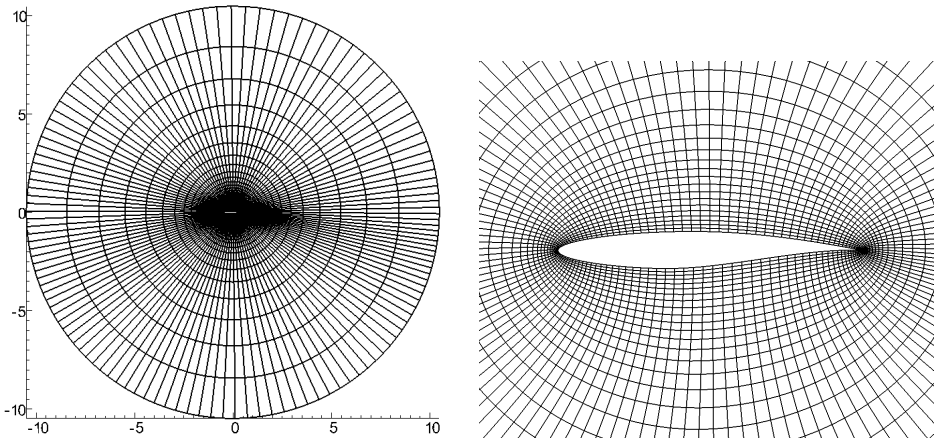
The above test case and the following case of flow around an asymmetric NACA-2822 airfoil were constructed as worse case examples for the AMG algorithm (in its current form). Although the performance is normally very efficient when the circulation is "frozen" (for instance when $\alpha = 0°$), as the circulation is updated it severely violates the coarse-level correction process. As a consequence, the V-cycle convergence becomes extremely slow. As already mentioned, there is much room for optimizing the AMG algorithm for this very particular problem, for instance by further improving the circulation extrapolation process or modifying the FMG approach to obtain a better initial guess for the fine-level.

12.7 Nonsymmetric airfoil NACA–2822

In this problem we take one more step further and consider a nonsymmetric airfoil. First, the nonsymmetry results in an irregular stretched grid near the airfoil surface. Second, such an airfoil generates lift even at a zero angle of attack. Therefore, the circulation must be calculated and the ghost cells above and below the cut must be updated after each smoothing sweep in order to satisfy the Kutta condition. Solutions to the NACA-2822 airfoil were attained on an O-type grid that was also constructed by the algorithm of Hawken [95]. The grid has 48 cells in the normal direction and 128 cells around the airfoil, while the coordinates are shown in Table 29. In the first example (NACA-0012), the farfield boundary is located ten chord lengths away from the airfoil. The grid and a magnified view of the grid near the airfoil's surface are presented in Figure 56.

Table 29: Coordinates of the NACA-2822 airfoil.

$x, [m]$	$y, [m]$	$x, [m]$	$y, [m]$
0.5	0	-0.4937678	0.02016554
0.4966792	0.00018965	-0.4740583	0.03022682
0.4852147	0.0007128	-0.4394766	0.04037793
0.4607264	0.00121574	-0.3878135	0.04987806
0.416639	0.00020189	-0.3171328	0.05759262
0.3473771	-0.00489418	-0.2271611	0.06215686
0.2517068	-0.01606706	-0.1202867	0.06208281
0.1348081	-0.03270936	-0.00204013	0.05554108
0.00738226	-0.04972611	0.1190656	0.0437382
-0.1177091	-0.05888357	0.2328184	0.03025188
-0.2292303	-0.05675405	0.3299344	0.01812037
-0.3206813	-0.04944643	0.4042574	0.00908261
-0.3905658	-0.04028333	0.4542647	0.0034868
-0.4406526	-0.03049537	0.4830459	0.00076342
-0.4739683	-0.02053009	0.4963978	0
-0.4932659	-0.01053022	0.5	-0.00055668
-0.5	0.01010243	–	–



(a) (b)

Figure 56: Structured curvilinear body-fitted grid of the O-type for NACA-2822. a) The mesh used for the NACA-2822 airfoil flow case study. b) A close-up of the grid near the airfoil.

The same boundary conditions were applied as for the NACA-0012 test case. This problem was solved for the cases of four different free-stream Mach numbers with a zero angle of attack. The Mach contours for four different Mach numbers are presented in Figure 57. Figures $a-c$ depicts subsonic cases ($M_\infty \leq 0.5$), while the fourth figure is at transonic conditions. At $M_\infty = 0.75$ a shock is present in the flow above the upper surface of the airfoil as can be seen in Figure 57(d).

Figure 58 shows the C_p distribution for the airfoil at $M_\infty = 0.1, 0.4, 0.5, 0.75$ and $\alpha = 0°$. The pressure coefficient is negative for pressure values less than the free-stream, which occurs on the top of the airfoil. In plots of the pressure coefficients for airfoils, the negative of the pressure coefficient is usually plotted to indicate that the lower pressure region is on the top of the airfoil and the high pressure region is on bottom of the airfoil.

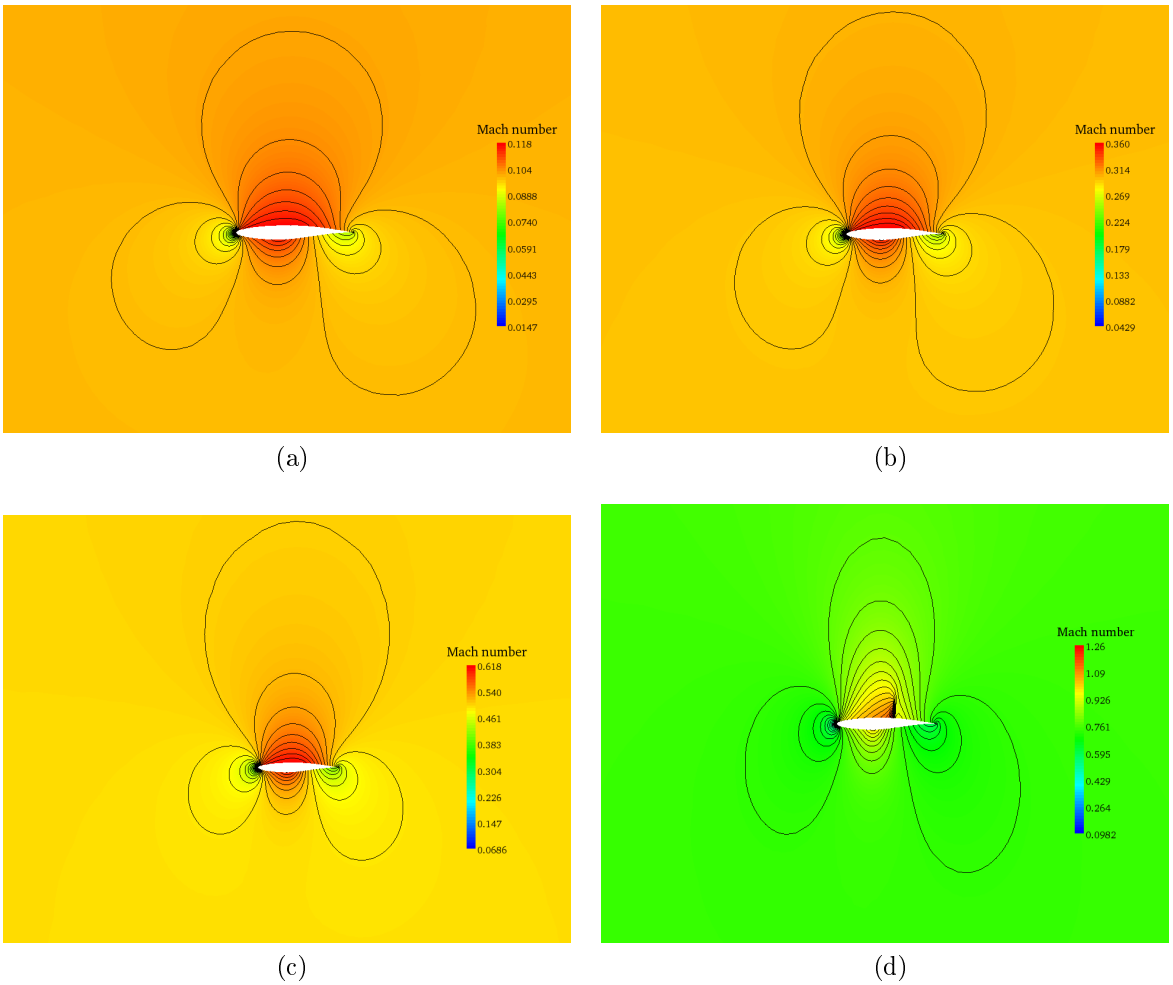


Figure 57: Distribution of velocity as computed on the (128×48) mesh, for different incident Mach numbers. a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.75$. Observe the shock appearing at the top and bottom surfaces.

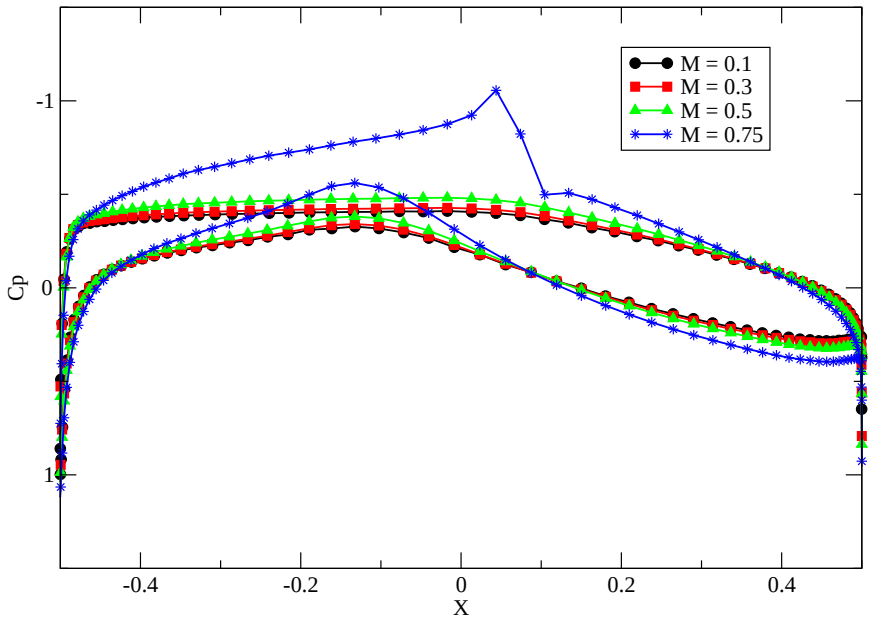


Figure 58: The distribution of pressure coefficient as computed using (128×48) mesh, for different Mach numbers at $\alpha = 0°$.

Table 30: Results of the circulation Γ for four cases of Mach number.

Circulation	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.5$	$M_\infty = 0.75$
Γ	1.221e-02	3.831e-02	7.097e-02	0.159

12.7.1 AMG performance

The first coarse-level for each case described above is sketched in Figure 59. When the flow is practically incompressible, the discrete operator is nearly isotropic and the points that comprise the coarse-level are distributed uniformly, as can be seen in Figure 59(a).

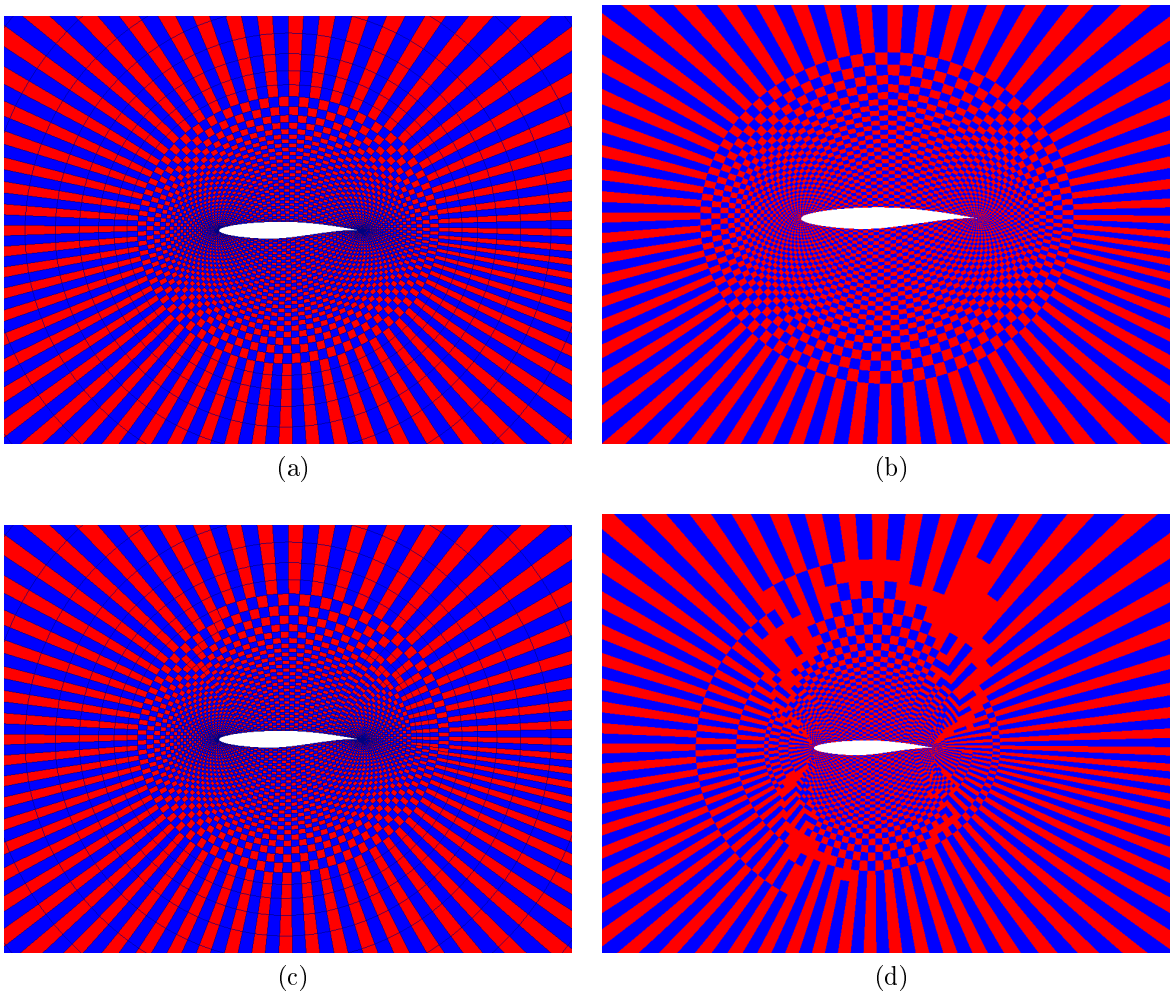


Figure 59: A magnified view of the coarsening area around the airfoil. Fine- and first coarse-level of the airfoil with $\alpha = 0°$ for four cases of Mach number: a) $M_\infty = 0.1$, b) $M_\infty = 0.3$, c) $M_\infty = 0.5$, d) $M_\infty = 0.75$. A blue point indicates an F-point and a red point indicates a C-point.

Table 31 presents the discrete L_2-norm of the residual and the convergence rate for each case of Mach number described above. The residual norm decreased rapidly for 10 to 12 V-cycles with the value in the corresponding $\|R^m\|_2$ column, while in the last few cycles a constant value was reached (asymptotic convergence rate). In the first three cases of $M_\infty = 0.1$, $M_\infty = 0.3$, and $M_\infty = 0.5$, the setup phase was implemented only once, followed by 12 V-cycles. Solving the problem on reduced resolution maintained the structure of the fine-scale problem and the convergence factor remained nicely bounded independent of the problem size. In the transonic case of $M_\infty = 0.75$, where the compressibility became significant, the results were obtained by repeating the setup phase 6 times while applying two V-cycles between each update. This resulted in a total of 12 V-cycles until the residual decreased to the desired level of $\tilde{}10^{-10}$. The transonic case provides a more difficult test for the algorithm. The convergence rate in the first three cycles is slightly damaged mainly due to the dominant nonlinearity. Although the first

three cycles are slow to converge, in the remaining V-cycles the convergence rate is lower than 0.17 without any the extra relaxation sweeps on the fine-level.

The grid complexity and operator complexity are presented in Table 32. It is clear that when the flow is subsonic the complexities are bounded. In the case of $M_\infty = 0.75$, the complexities are somewhat high, relative to our above requirements, while the reasons for the increased complexities are similar to those stated in the previous problems (for instance channel with a bump). A possible way to improve the complexities is by aggressive coarsening. This approach was implemented but, as expected, the convergence became considerably slower (above 0.3). Aggressive coarsening not only causes the smoothing to be less effective but also interpolation is significantly less accurate.

The convergence histories for the five cases are described in Figure 60. The effect of the Mach number on the AMG performance is clearly visible. The case of $M_\infty = 0.75$ results in a supersonic flow regime (in the upper surface) that is terminated by a shock wave. Convergence is somewhat slower here. It requires nearly 12 V-cycles to decrease the L_2-norm of the residual to a level of 10^{-10}, which is twice the number of AMG iterations required in the case of $M_\infty = 0.1$.

Table 31: The results of AMG V-cycles applied to the flow around a NACA-2822. The second norm of the residual $\|R^m\|_2$ and the convergence rate C_f after each V-cycle are presented for mesh size 128×48.

	$M_\infty = 0.1$		$M_\infty = 0.3$		$M_\infty = 0.5$		$M_\infty = 0.75$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	5.649e-03	–	6.100e-02	–	0.102e-00	–	8.442e-02	–
1	6.026e-04	0.10	1.850e-02	0.30	2.445e-02	0.03	5.955e-02	0.70
2	6.644e-05	0.11	1.622e-03	0.09	2.668e-03	0.11	8.632e-03	0.14
3	5.636e-06	0.08	1.469e-04	0.09	2.116e-04	0.08	1.459e-03	0.17
4	4.483e-07	0.08	1.330e-05	0.09	1.898e-05	0.09	2.469e-04	0.17
5	3.460e-08	0.08	1.195e-06	0.09	1.594e-06	0.08	3.680e-05	0.15
6	2.680e-09	0.08	1.072e-07	0.09	1.591e-07	0.10	5.388e-06	0.15
7	2.093e-10	0.08	9.687e-09	0.09	1.533e-08	0.10	7.915e-07	0.15
8	1.658e-11	0.08	8.748e-10	0.09	1.200e-09	0.08	1.207e-07	0.15
9	1.330e-12	0.08	7.891e-11	0.09	9.192e-11	0.08	1.945e-08	0.16
10	1.051e-13	0.08	7.076e-12	0.09	1.041e-11	0.11	3.354e-09	0.17
11	1.580e-14	0.15	5.900e-13	0.08	1.128e-12	0.11	5.695e-10	0.17
12	1.515e-14	0.96	7.421e-14	0.13	6.260e-14	0.06	9.681e-11	0.17

Table 32: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

complexities	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.5$	$M_\infty = 0.75$
C_Ω	1.90	1.92	1.96	2.08
C_L	3.03	2.94	3.02	4.95

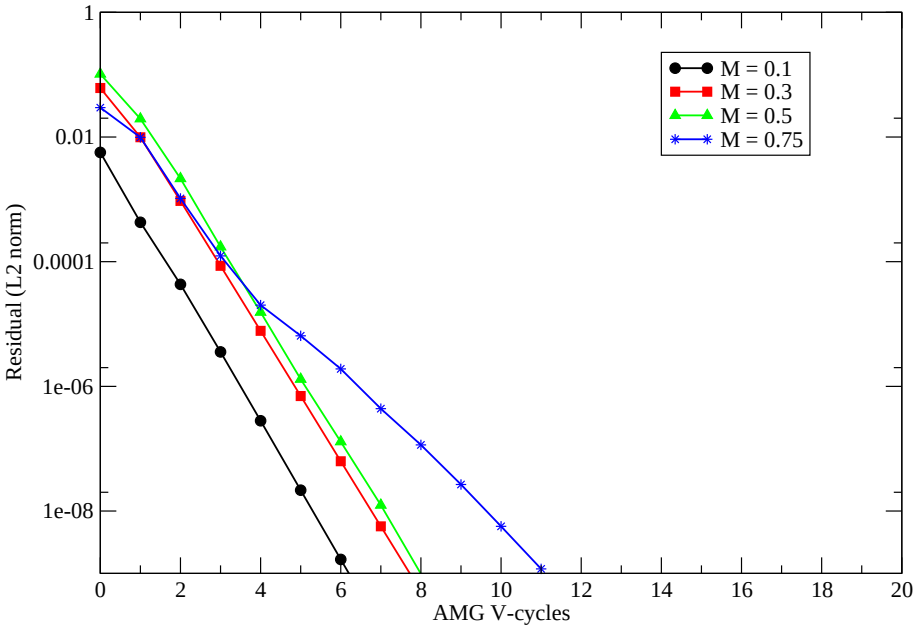


Figure 60: Convergence histories.

This problem was solved for two more grid sizes of 64×24 and 32×12. Convergence factor C_f and L_2-norm of the residual $\|R^m\|_2$ are presented in Table 33. We observe that for subsonic flow the AMG exhibits the same type of convergence that was observed on the finer grid. The residual L_2-norm is decreased by a relatively constant factor, less than an order of magnitude per each V-cycle. It takes about 8 V-cycles to reach a residual of 10^{-12}. As the Mach number increases the convergence degrades, albeit we find it still very efficient for this problem in light of the strong irregularity and anisotropy. The storage requirements are larger than that for the subsonic (isotropic case) since AMG performs one-dimensional coarsening in the azimuth direction at the farfield since the strong connection in the ξ-direction (azimuthal direction) arises from a large grid spacing in η- direction. This fact has an adverse effect on the complexities.

Table 33: The results of AMG V-cycles applied to the flow around NACA-2822 with $\alpha = 0°$. The second norm of the residual $\|R^m\|_2$ after each V-cycle and the convergence factor C_f are detailed for two mesh sizes, 32×12 and 64×24.

	$M_\infty = 0.1$				$M_\infty = 0.3$			
grid size →	32×12		64×24		32×12		64×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	1.328e-03	–	1.114e-02	–	4.288e-03	–	3.296e-02	–
1	1.004e-04	0.07	9.747e-04	0.08	5.875e-04	0.13	6.170e-03	0.18
2	7.135e-06	0.07	9.205e-05	0.09	5.933e-05	0.10	4.919e-04	0.08
3	4.870e-07	0.07	7.371e-06	0.07	5.348e-06	0.09	3.398e-05	0.07
4	3.277e-08	0.07	5.523e-07	0.07	4.673e-07	0.09	2.294e-06	0.07
5	2.213e-09	0.07	4.058e-08	0.07	4.059e-08	0.09	1.552e-07	0.07
6	1.510e-10	0.07	2.974e-09	0.07	3.521e-09	0.09	1.054e-08	0.07
7	1.041e-11	0.07	2.186e-10	0.07	3.052e-10	0.09	7.190e-10	0.07
8	7.247e-13	0.07	1.613e-11	0.07	2.640e-11	0.09	4.921e-11	0.07
9	5.064e-14	0.07	1.194e-12	0.07	2.277e-12	0.09	3.384e-12	0.07
10	3.700e-15	0.07	8.842e-14	0.07	1.963e-13	0.09	2.333e-13	0.07
11	7.061e-16	0.19	6.497e-15	0.07	1.703e-14	0.09	1.696e-14	0.07
12	5.735e-16	0.81	1.619e-15	0.25	2.390e-15	0.14	5.257e-15	0.07
C_Ω	**1.85**		**1.90**		**1.89**		**1.91**	
C_L	**2.54**		**2.91**		**2.53**		**2.82**	

	$M_\infty = 0.5$				$M_\infty = 0.75$			
grid size →	32×12		64×24		32×12		64×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	8.552e-03	–	7.200e-02	–	5.126e-02	–	8.727e-02	–
1	2.801e-03	0.32	5.950e-02	0.82	2.350e-02	0.45	4.070e-02	0.46
2	2.861e-04	0.10	6.592e-03	0.11	6.902e-03	0.29	2.065e-03	0.05
3	2.719e-05	0.10	7.183e-04	0.11	2.262e-03	0.33	2.500e-04	0.12
4	2.511e-06	0.09	7.851e-05	0.11	1.746e-04	0.08	1.467e-05	0.06
5	2.294e-07	0.09	8.533e-06	0.11	1.155e-05	0.07	2.934e-06	0.20
6	2.085e-08	0.09	9.283e-07	0.11	7.904e-07	0.07	4.916e-07	0.17
7	1.886e-09	0.09	1.013e-07	0.11	4.775e-08	0.06	8.675e-08	0.18
8	1.700e-10	0.09	1.105e-08	0.11	5.469e-09	0.11	1.339e-08	0.15
9	1.527e-11	0.09	1.206e-09	0.11	4.527e-10	0.08	1.897e-09	0.14
10	1.355e-12	0.09	1.316e-10	0.11	4.591e-11	0.10	2.564e-10	0.14
11	1.084e-13	0.08	1.439e-11	0.11	7.794e-12	0.17	3.513e-11	0.14
12	8.545e-15	0.08	1.597e-12	0.11	4.771e-12	0.61	6.247e-12	0.18
C_Ω	**1.91**		**1.92**		**1.98**		**2.02**	
C_L	**2.63**		**2.82**		**3.94**		**4.60**	

Table 34 shows the FMG performance applied to the flow around an NACA-2822 airfoil with $\alpha = 0°$ for cases of four different free-stream Mach numbers with mesh size of 128×48 cells. The row labeled "FMG(1,1), number of levels" lists how many coarse-levels (geometrically coarsened $h \to 2h$) were needed in the FMG

scheme. The row labeled "AMG V(1,1) cycles" gives the number of AMG V-cycles used until a desired level of residual was reached in each coarse-level. Also the average convergence factor is presented for each level. In the case of $M_\infty < 0.5$, five AMG V-cycles were applied in each coarse-level as part of the FMG algorithm. Observe that the average convergence rate were around an order of magnitude per cycle. It was clear that when the flow is subsonic through all the domains, five V-cycles were sufficient for getting a reasonable correction on the fine-level. In the transonic case ten V-cycles were applied in each coarse-levels.

Table 34: The FMG performance applied to the flow around an airfoil NACA-2822 with $\alpha = 0°$ for four Mach numbers. The FMG(1,1) scheme has one SGS relaxation on the descent phase and one SGS relaxation on the ascent phase. The FMG(1.1) number of levels is presented in the first row. The second row includes the AMG V-cycles performed in the FMG process. The average convergence factor between successive AMG V-cycles is presented in the third row.

	$M_\infty = 0.1$			$M_\infty = 0.3$		
grid size $\longrightarrow$	32×12	64×24	128×48	32×12	64×24	128×48
FMG(1,1), number of levels	3	4	4	3	4	4
AMG V(1,1) - cycles	1	5	5	1	5	5
Average convergence factor,C_f	–	0.09	0.10	–	0.09	0.10
	$M_\infty = 0.5$			$M_\infty = 0.75$		
FMG(1,1), number of levels	3	4	4	3	4	4
AMG V(1,1) - cycles	1	5	5	1	10	10
Average convergence factor,C_f	–	0.11	0.12	–	0.12	0.12

12.8 Transonic diffuser

The purpose of this test-case is to evaluate the AMG performance while simulating the operation of a converging-diverging diffuser, which is an important and basic component associated with propulsion and the high speed flow of gases. This application often places strong demands on CFD algorithms. For example, one purpose of the diffuser is to decelerate the flow ahead of the engine. Another purpose is to connect the inlet with the engine. In many cases, the inlet and engine axes are offset, and the diffuser must turn the flow [14]. In this exercise, both subsonic and transonic flow through a converging-diverging diffuser are investigated. Mach number variation and shock formation may be examined. This flow problem can be found practically in any gas dynamics textbook for example [15, 14].

12.8.1 Problem definition and boundary condition

The diffuser has a rectangular cross-section. It has a flat bottom wall and a converging-diverging channel with a maximum 10° divergence angle at the top wall. A diagram of the diffuser model is shown in Figure 61. The coordinates of the transonic diffuser are presented in Figure 35. The ratio of the inlet area to

the throat area is $\frac{A_{in}}{A_{throat}} = 1.4114$ and the ratio of the exit area and the throat area is $\frac{A_{out}}{A_{throat}} = 1.5$. This problem was solved on grids with three different levels of resolutions in order to check the AMG's scalability. These grids are as follows: 120×40, 60×20, and 30×10, while the finer mesh is presented in Figure 62. The grid is clustered close to upper and lower walls and in the throat (horizontally), with a stretching factor of 1.1.

Table 35: Coordinate of the transonic diffuser.

$x, [m]$	$y, [m]$	$x, [m]$	$y, [m]$
-4.04000	0	2.03921	1.08010
8.65000	0	2.11764	1.08713
-4.04000	1.49986	2.19608	1.09446
-3.75143	1.41140	2.27451	1.10208
-3.46286	1.41140	2.35294	1.10998
-3.17429	1.41140	2.43137	1.11816
-2.88572	1.41140	2.50980	1.12661
-2.59714	1.41140	2.58824	1.13531
-2.30857	1.41140	2.66667	1.14425
-2.02082	1.38302	2.74510	1.15343
-1.73635	1.28072	2.82353	1.16282
-1.45928	1.17730	2.90196	1.17242
-1.19451	1.10234	2.98039	1.18220
-0.94667	1.05538	3.05882	1.19215
-0.71915	1.02827	3.13725	1.20226
-0.51364	1.01346	3.21569	1.21249
-0.33014	1.00577	3.29412	1.22284
-0.16727	1.00204	3.37255	1.23329
-0.02270	1.00046	3.45188	1.24381
0.10619	1.00001	3.53484	1.25449
0.22237	1.00019	3.62593	1.26570
0.32846	1.00083	3.73060	1.27801
0.42677	1.00184	3.85392	1.29211
0.51926	1.00316	3.99963	1.30860
0.60748	1.00475	4.16959	1.32777
0.69263	1.00658	4.36380	1.34954
0.77560	1.00866	4.58076	1.37333
0.85708	1.01099	4.81792	1.39815
0.93752	1.01357	5.07227	1.42268
1.01727	1.01642	5.34067	1.44548
1.09656	1.01953	5.62020	1.46514
1.17554	1.02293	5.90828	1.48057
1.25432	1.02661	6.20279	1.49121
1.33297	1.03058	6.50201	1.49720
1.41154	1.03486	6.80461	1.49956
1.49006	1.03943	7.10959	1.49999
1.56854	1.04432	7.41623	1.50000
1.64701	1.04951	8.03242	1.50000
1.72546	1.05501	8.34121	1.50000
1.80390	1.06082	8.65000	1.50000
1.88234	1.06694	–	–
1.96078	1.07336	–	–

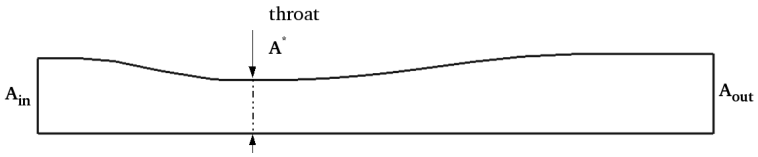


Figure 61: A diagram of the transonic diffuser.

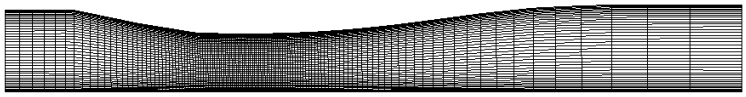


Figure 62: Mesh used for the flow through the transonic diffuser.

The following boundary conditions were specified:

- Subsonic flow at the inlet and the mass flow rate at the outlet are the same, in the x-direction.
- Solid-wall boundary condition at the top and bottom of the diffuser.

12.8.2 Qualitative results

The overall characteristics of flow field were examined to make sure that the solution reflected the expected behavior. Figure 63 presents cases of subsonic and transonic flow. Figure 63(a) shows the flow through the diffuser when it is completely subsonic. The flow accelerates out of the chamber through the converging section, reaching its maximum speed at the throat. The flow then decelerates through the diverging section. Figure 63(d) presents the case where the Mach number at the inlet is increased to $M_\infty = 0.46$. The flow at the throat reaches the sonic speed (choked throat). Unlike the subsonic flow, the supersonic flow accelerates as the cross-section area is increased. The region of the supersonic acceleration is terminated by a normal shock wave, as can be seen in Figure 63(d).

In order to verify the simulation results, in the subsonic region where the flow is isentropic, the relation between the geometry and the flow properties is needed. The flow of fluid in the transonic-diffuser can be modeled by assuming isentropic flow. The ideal gas assumption relates density, pressure, and temperature,

$$p = \rho RT. \tag{200}$$

Isentropic flow, no change in entropy, is possible if no heat transfer or friction occurs. The properties γ and R fully specify an ideal gas. Sonic velocity of an ideal gas is a function of gas properties and temperature, thus,

$$M = \frac{V}{c} = \frac{V}{\sqrt{\gamma RT}}. \tag{201}$$

With these assumptions of ideal gas and isentropic flow, the ratios of pressure, density, and temperature can be related to their stagnation values at a given Mach number by the following relations:

$$\begin{aligned}
\frac{P_0}{P} &= \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}}, \\
\frac{T_0}{T} &= \left(1 + \frac{\gamma - 1}{2} M^2\right), \\
\frac{\rho_0}{\rho} &= \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{1}{\gamma-1}}.
\end{aligned} \tag{202}$$

The mass conservation is given by

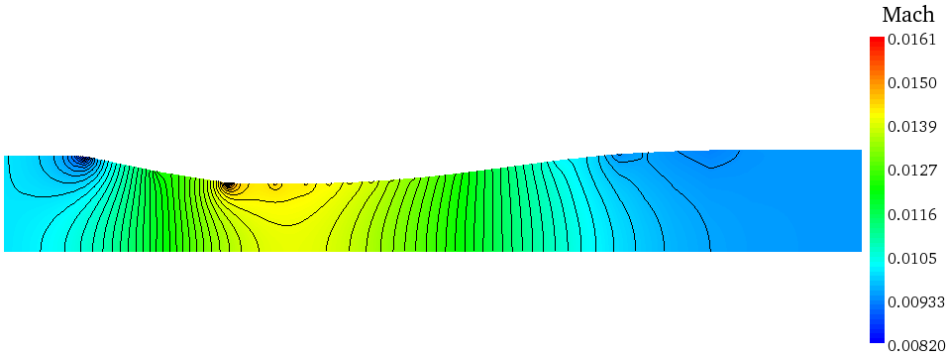
$$\rho A V = \rho^* A^* V^* = const. \tag{203}$$

This equation states that the mass flowing through the nozzle must be constant. The superscript "star" signifies critical conditions (when the flow reaches a sonic speed), where the Mach number is unity at the throat. It is possible to express the area ratio A/A^* in terms of the Mach number of the flow,

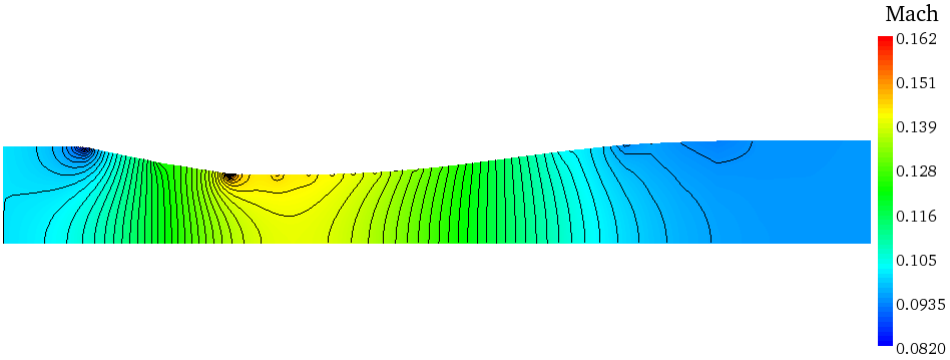
$$\frac{A}{A^*} = \frac{1}{M_\infty} \left[\frac{1 + \frac{\gamma-1}{2} M_\infty^2}{1 + \frac{\gamma-1}{2}} \right]^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{204}$$

According to Eq. (204) when the inlet Mach number is $M_\infty = 0.46$, as computed in Figure 63(d), Eq. (204) becomes 2.01, which is exactly the ratio between the inlet area and the throat area. Thus, there is a good agreement between the nozzle theory calculations and the numerical simulation.

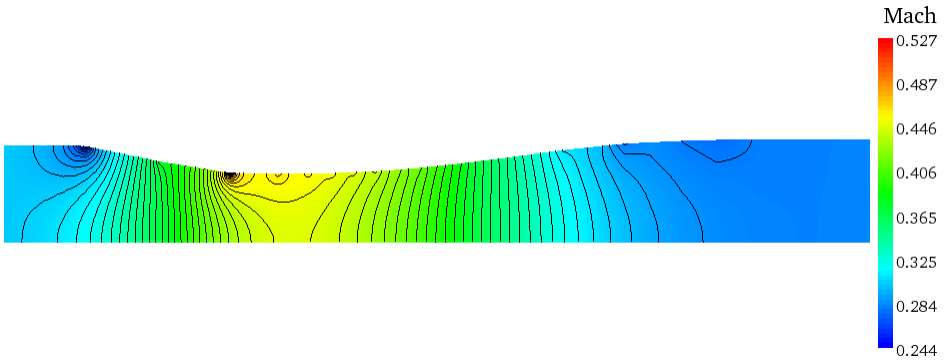
A pressure coefficient along the diffuser upper wall for four cases of inlet velocity is presented in Figure 64. No significant differences are observed between the results obtained with Mach number ranging from 0.01 to 0.3. A strong pressure gradient is present in the case of $M_\infty = 0.46$ where a shock is obtained at the diverging section of the diffuser.



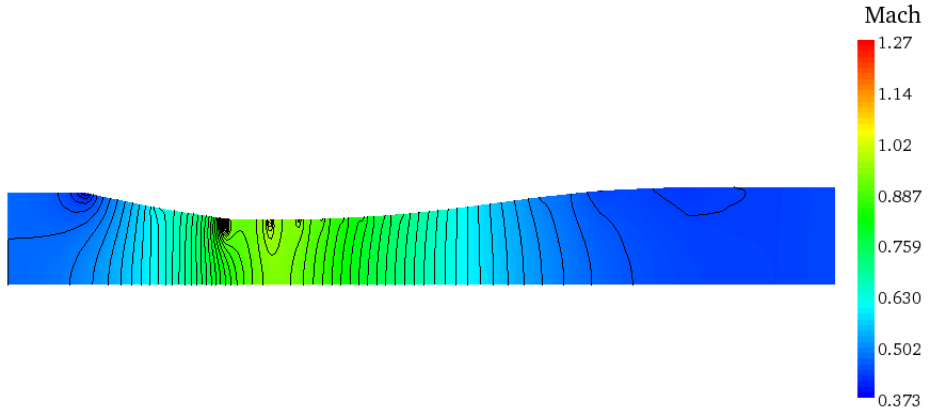
(a)



(b)



(c)



(d)

Figure 63: Mach number isolines as computed on the (120×40) mesh, for an inlet Mach number: a) $M_\infty = 0.01$, b) $M_\infty = 0.1$, c) $M_\infty = 0.2$, d) $M_\infty = 0.3$, c) $M_\infty = 0.46$. Fifty contours were sketched between the maximum and minimum presented Mach numbers. Observe the shock wave in the fourth case.

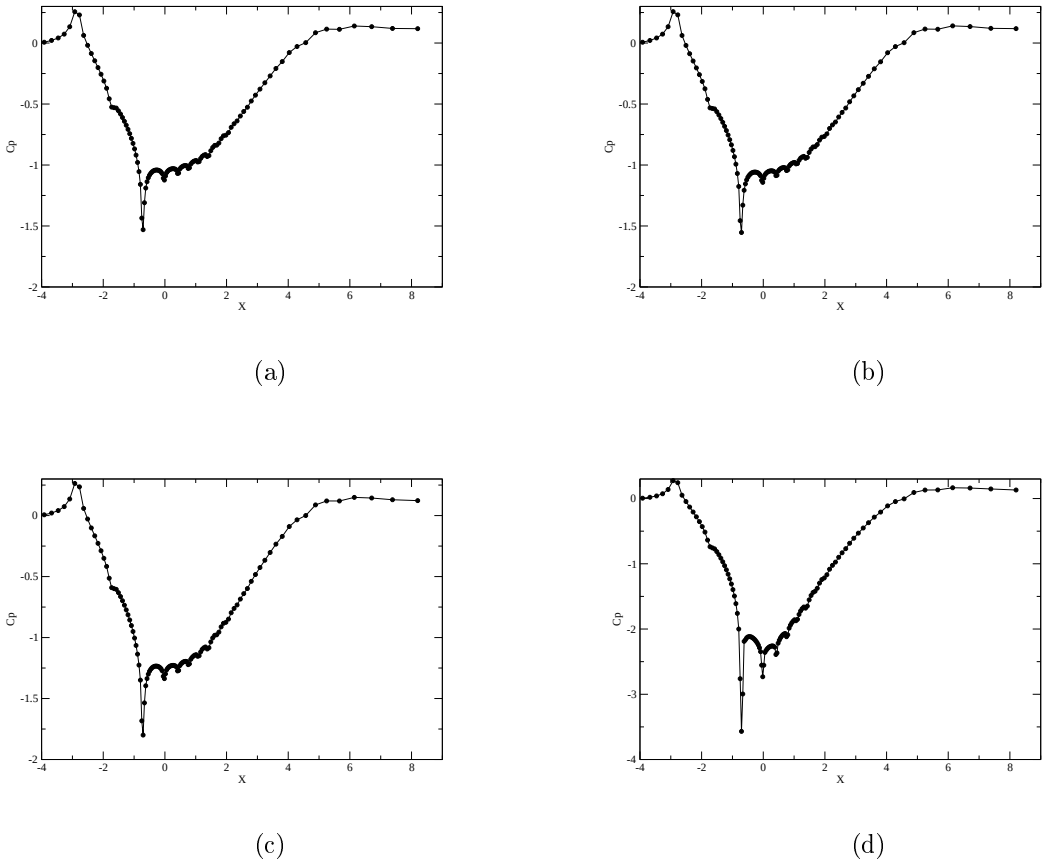


Figure 64: The pressure coefficient calculated on the upper wall using the (121×41) mesh size for the above flow conditions. Observe the shock appearing when $M_\infty = 0.465$.

12.8.3 AMG performance

So far we have considered only symmetric problems. Practical experience has shown that no particular problems were caused while solving a nonsymmetric problem. The main reason for that is the way we construct the coarse-level matrix. Using the coarsening process while using a direct approximation (detailed in Section 8) rather than Galerkin operator, we are not limited in the type of resultant coarse-level matrix that is obtained – for instance, whether the matrices are positive definite, weakly diagonally dominant, or a various class of matrices that are not in the M-matrix form (with large negative entries off-diagonal). In the following example and the next one (flow through a nozzle), AMG turned out to yield robust and fast convergence.

The first coarse-level for each case described above is sketched in Figure 65. When the flow is incompressible ($M = 0.01$), the operator is nearly isotropic and is constructed of a nearly 17-point stencil (most of the points "far away" from the

point i are very small, so this operator "behaves" as a nine-point stencil). However, since the grid is highly stretched near the bottom and upper walls and also in the throat area, this irregularity affects the coarse points selection, as can be seen in Figure 65(a). It can be seen that a red-black coarsening is obtained through most of the domain except at the convergent and divergent parts of the throat, where two kinds of coarsening patterns were obtained. In the throat (where the area is minimal) the cell's aspect ratio is nearly unity; therefore, the coarsening algorithm distributes the coarse points uniformly. In the convergent and divergent parts of the diffuser, where the anisotropy is largest, the grid is coarsened in the direction of strong connection, in the y-direction. A significant contribution to the strong connection in the y-direction is the irregularity of the grid in these areas. As the velocity is increased the equation becomes anisotropic, and in the case of $M_\infty = 0.46$ the dynamic threshold was applied in order to reach convergence. Since the coarsening process while using the dynamic threshold yields larger coarse operators (since more coarse-points are selected), this phenomenon is well reflected in the coarsening pattern presented in Figure 65(d). However, except for the uniform coarsening in the throat area, the coarsening pattern is essentially the same.

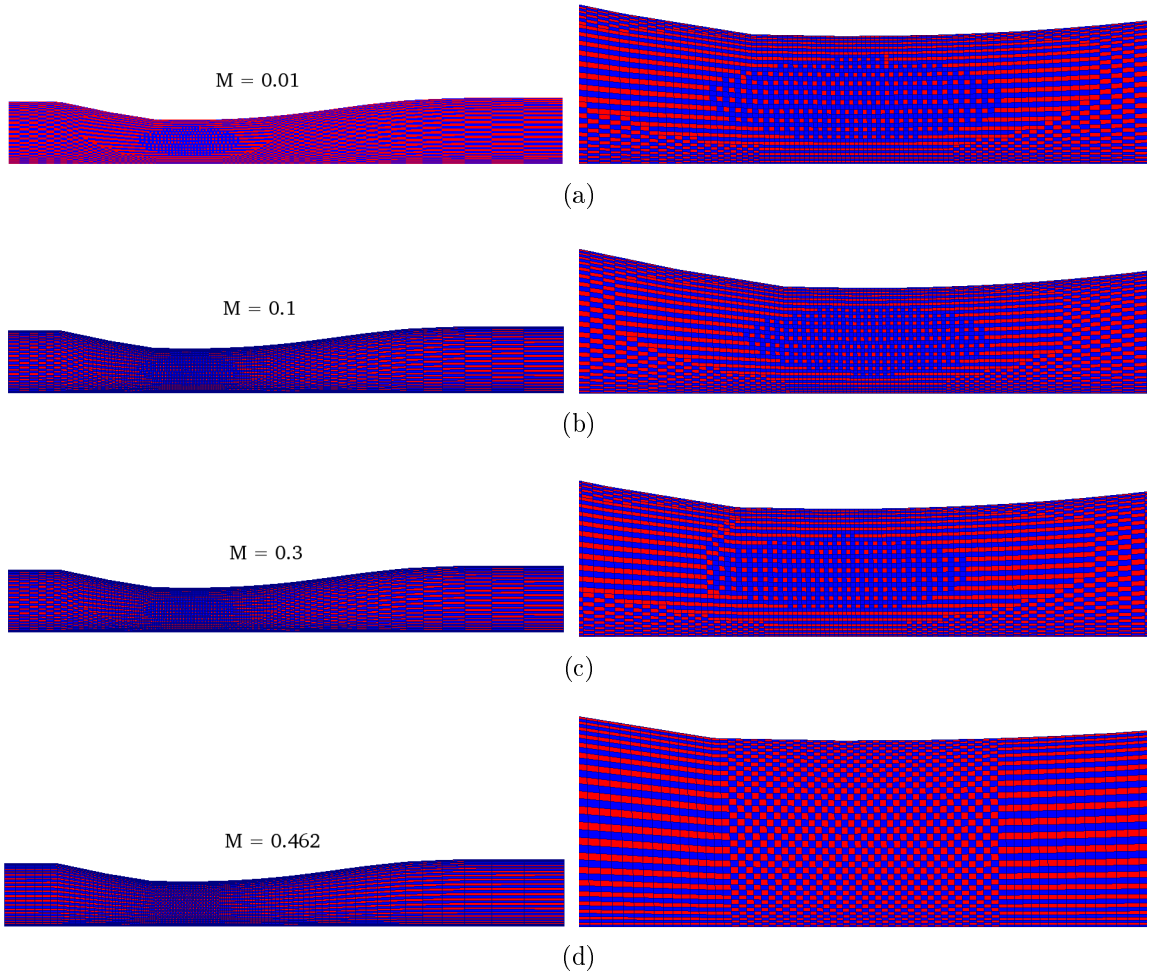


Figure 65: The finest and the first coarse-level for mesh size of (121×41). The red color corresponds to the C-point and the blue color corresponds to the F-point. The four test cases are solved with various flow conditions as follows: a) $M_\infty = 0.01$, b) $M_\infty = 0.1$, c) $M_\infty = 0.3$, d) $M_\infty = 0.462$. The pictures on the right are magnified views of the coarsening pattern around the throat area.

Table 36 presents the discrete L_2-norm of the residual for each case of Mach number described above. Both the Mach contours and the convergence history are evidence that the AMG solves the problem to the level of discretization on each grid. The residual norm decreased rapidly for 8 to 10 V-cycles with the value in the corresponding $\|R^m\|_2$ column, while in the last few cycles a constant value was reached (asymptotic convergence rate). In the first two cases of $M_\infty = 0.01$ and $M_\infty = 0.1$, the setup phase was implemented only once, followed by 8 V-cycles. Solving the problem on reduced resolutions, the structure of the fine-scale problem remains very similar and so does the convergence factor, verifying that the solver's efficiency is independent on the problem size.. In the case of $M_\infty = 0.3$, where the compressibility became significant, the results were obtained by repeating the setup phase six times while applying two V-cycles between each update. This results in a total of 8 V-cycles until the residual decreased to the desired level of $\tilde{}10^{-10}$. The transonic case, $M_\infty = 0.46$, provides a more difficult test for the

algorithm. In this case the setup phase was repeated 6 times resulting in 12 V-cycles. The convergence rate in the first three cycles was deteriorated slightly. Although the first three cycles are slower to converge, in the remaining V-cycles the convergence rate is lower than 0.1 without any local smoothing sweeps around the shock waves.

The convergence histories for the five cases are depicted in Figure 66. The effect of the Mach number on the AMG performance is clearly shown. In the first three cases, when the flow is subsonic, it takes around 8 cycles to reduce the L_2-norm of the residual to a level of 10^{-10}. The case of $M_\infty = 0.46$ results in a supersonic flow regime that is terminated by a shock wave. Convergence is somewhat slower here. It requires nearly 12 multigrid cycles to decrease the L_2-norm of the residual to a level of 10^{-10}.

The grid complexity and operator complexity are presented in Table 37. It is clear that when the flow is subsonic the complexities are bounded. In the case of $M_\infty = 0.46$ an operator complexity of $C_L = 3.80$ is too high, relative to our requirements, while the reasons for the increased complexities are similar to those stated in the previous problems. Two possible ways to improve the complexities for the transonic case ($M_\infty = 0.46$) are by more aggressive coarsening or truncation of the coarse-level operator.

Table 36: The results of AMG V-cycles applied to the flow through the transonic convergent-divergent diffuser. The second norm of the residual $\|R^m\|_2$ and the convergence factor C_f are presented for each AMG V-cycle. The mesh size 120×40.

	$M_\infty = 0.01$		$M_\infty = 0.1$		$M_\infty = 0.3$		$M_\infty = 0.46$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	4.933e-04	–	4.947e-03	–	1.366e-02	–	1.861e-02	–
1	1.120e-04	0.22	1.140e-03	0.23	6.865e-03	0.50	3.588e-03	0.19
2	5.680e-06	0.05	5.566e-05	0.05	3.771e-04	0.05	1.364e-03	0.38
3	3.859e-07	0.07	3.382e-06	0.06	3.187e-05	0.08	5.269e-04	0.39
4	2.778e-08	0.07	2.050e-07	0.06	2.799e-06	0.09	3.287e-04	0.62
5	2.102e-09	0.08	1.307e-08	0.06	1.293e-07	0.05	2.177e-05	0.07
6	1.609e-10	0.08	8.361e-10	0.06	8.572e-09	0.07	1.634e-06	0.08
7	1.242e-11	0.08	5.359e-11	0.06	8.630e-10	0.10	1.834e-07	0.11
8	9.620e-13	0.08	3.395e-12	0.06	2.682e-11	0.03	1.905e-08	0.10
9	7.469e-14	0.08	2.157e-13	0.06	4.277e-12	0.16	1.798e-09	0.09
10	6.379e-15	0.09	3.707e-14	0.17	3.189e-13	0.07	2.258e-10	0.13
11	2.832e-15	0.44	3.088e-14	0.83	1.283e-13	0.40	1.966e-11	0.09
12	2.607e-15	0.92	3.082e-14	1.00	1.342e-13	1.05	2.260e-12	0.11
13	–	–	–	–	–	–	3.300e-13	0.15
14	–	–	–	–	–	–	2.285e-13	0.69

Complexities	$M_\infty = 0.01$	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.46$
C_Ω	1.74	1.74	1.77	2.03
C_L	2.44	2.48	2.54	3.80

Table 37: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

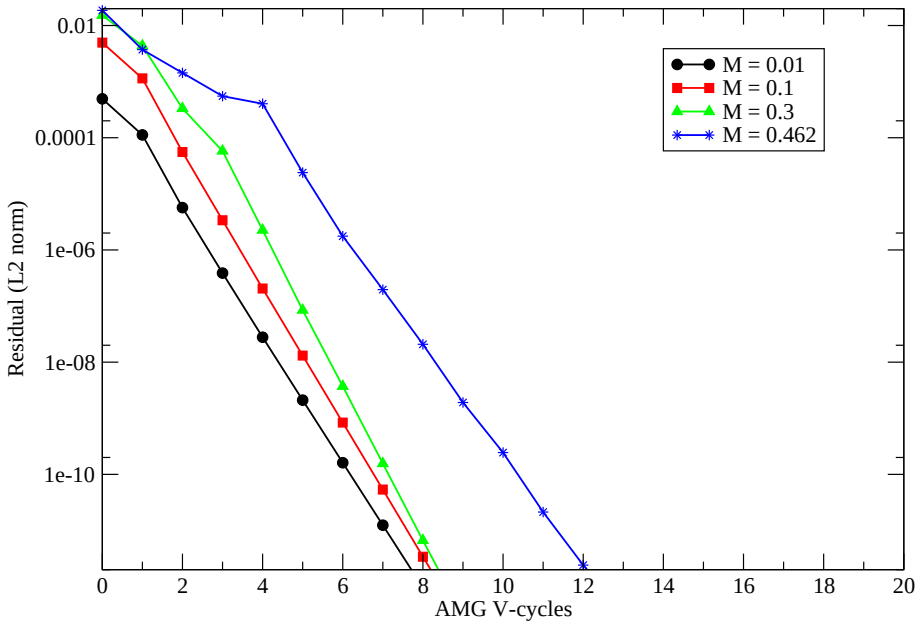


Figure 66: Convergence histories.

Table 38 shows the average number of rows and number of nonzeros for the above four cases of $M_\infty = 0.01 - 0.46$. Several comments are in order here. The problems were solved with maximum 6 levels. Fixed threshold is applied to the first three cases while in the fourth case of $M_\infty = 0.46$ a dynamic threshold is applied. A second pass process is implemented only in the first coarse-level, for all the test cases. In the case of $M_\infty = 0.01$ the coarsening is relatively fast with a coarsening ratio of 0.45. This can be clearly seen in the coloring scheme described in Figure 65(a). Subsequent coarsening then becomes slightly faster because the coarse-level, which was produced by direct approximation (see Section 8), becomes larger on coarser levels. Also in this case, the number of nonzeros is decreased in the coarse-levels while the situation is different as the anisotropy is increased. In the case of $M_\infty = 0.3$, the first coarsening step tends to be relatively fast (0.5), in terms of reduction of grid points, while at the same time the size of the resulting

stencil on the second level becomes substantially larger than on the finest one. The situation is even more dramatic as the anisotropy is increased. The reason for the relatively slow coarsening for $M_\infty \geq 0.3$ is due to the increased stencil size in the first coarse-level and the dynamic threshold as a measure of strength between the matrix variables. The alternative to use a fixed threshold, particularly in this transonic case, would result in a much slower convergence rate, as already described above.

This problem was solved for two more cases of mesh consisting of 64×24 and 32×12 points. Both the L_2-norm of the residual and the convergence factor for two cases of $M_\infty = 0.1$ and $M_\infty = 0.46$ are presented in Table 39. The grid complexity and operator complexity for these cases are also presented in Table 39. As one can see, for both grid sizes the residual norm decreases by a relatively constant factor with each V-cycle. This continues until it levels off after about 12 V-cycles. Although each V-cycle gives good convergence, as the resolution is decreased one might expect a faster convergence and not slower as is presented for both cases of Mach numbers (see Table 36). In order to check if this degradation is a result of an error that is not sufficiently smooth, an extra relaxation was applied on the finest level, but the convergence rate was marginally improved. This result ensures that the main source of this degradation is the interpolation process, which becomes much less accurate when the grid is more and more "diluted". Another possible reason that contributes to the overall degradation in small scale problems is the stretching of the grid near the boundaries, which becomes more extreme as the the resolution is decreased. This situation is more dramatic when supersonic flow regions are involved, as can be seen in Table 39.

Also in this context, for $M_\infty = 0.46$ one can see that the operator complexity with grid size 60×20 is $C_L = 3.23$, which is slightly improved compared to $C_L = 3.80$ obtained in a grid size of 120×40. This change is mainly a result of the grid irregularity. As the resolution decreased, the coarse-level matrix contains many fewer nonzeros compared to the fine-level. For instance, in grid size 60×20 the first coarse-level contains 12.5 nonzeros more than the finest level (first coarse-level - 79365 nonzeros, finest level - 69424 nonzeros). For grid size 120×40 the first coarse-level contains 25% more nonzeros than the finest level (coarse-level - 108127, finest level - 83224 points). This increase of the coarse-level operator, at least in the first coarse-level, influenced the operator complexity substantially. In order to understand it intuitively, the best way is to look at the first coarse-levels for the above two resolutions, as sketched in Figure 65 for $M_\infty = 0.46$. In grid size 80×50, in the convergent area of the diffuser a line-wise coarsening is obtained in the y-direction. On the other hand, in the same region but with grid size 120×40 a red-black coarsening was obtained. As a consequence, since the final values of C_Ω and C_L are influenced by the size of the coarse-level operators, much larger values were obtained. For both grid sizes 120×40 and 80×50, except for the throat area, the coarsening pattern is essentially the same.

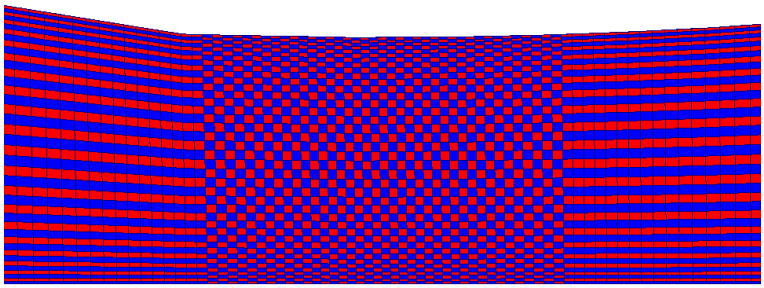
Table 38: The Results of AMG V-cycles applied to the flow through a transonic diffuser.

	Number of rows			
Level	$M_\infty = 0.01$	$M_\infty = 0.1$	$M_\infty = 0.3$	$M_\infty = 0.46$
A^m	4800	4800	4800	4800
A^{m+1}	2204	2203	2257	2400
A^{m+2}	897	907	1029	1321
A^{m+3}	337	347	435	859
A^{m+4}	125	126	177	534
A^{m+5}	–	–	66	315
A^{m+6}	–	–	–	188
Number of nonzeros				
A^m	83224	83224	83224	83224
A^{m+1}	67374	70068	100602	108127
A^{m+2}	34140	33831	48820	60688
A^{m+3}	14229	14740	22982	33948
A^{m+4}	4698	4813	7805	17736
A^{m+5}	–	–	1676	8836
A^{m+6}	–	–	–	3697
Average entries per row				
A^m	17.3	17.3	17.3	17.3
A^{m+1}	30.5	31.8	44.5	45
A^{m+2}	38	37.2	47.4	45.9
A^{m+3}	42.2	42.4	52.8	39.5
A^{m+4}	37.5	38.1	44	33.2
A^{m+5}	–	–	25.4	28
A^{m+6}	–	–	–	19.6

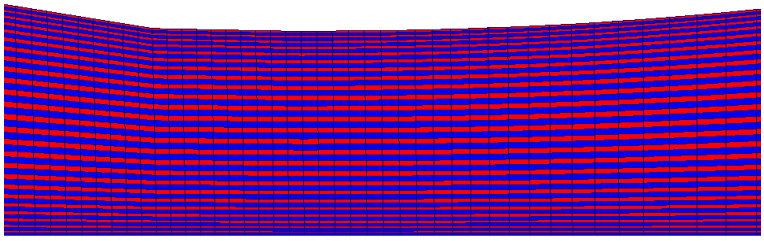
Table 39: The results of AMG V-cycles applied to the flow through the transonic diffuser. The discrete second norm of the residual $\|R^m\|_2$ after each V-cycle and the convergence factor C_f are presented for various Mach numbers and two grids of 32×12 and 64×24.

	$M_\infty = 0.01$				$M_\infty = 0.1$			
grid size $\longrightarrow$	30×10		60×20		30×10		60×20	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	2.226e-03	–	1.672e-03	–	2.227e-02	–	1.674e-02	–
1	2.837e-04	0.12	1.681e-04	0.10	2.859e-03	0.12	1.856e-03	0.11
2	1.442e-05	0.05	5.516e-06	0.03	1.450e-04	0.05	5.142e-05	0.03
3	8.765e-07	0.06	1.695e-07	0.03	9.540e-06	0.07	1.806e-06	0.04
4	5.857e-08	0.07	6.545e-09	0.04	6.834e-07	0.07	6.405e-08	0.04
5	4.259e-09	0.07	2.411e-10	0.04	5.254e-08	0.08	2.707e-09	0.04
6	3.384e-10	0.08	1.020e-11	0.04	4.352e-09	0.08	1.090e-10	0.04
7	2.912e-11	0.09	4.207e-13	0.04	3.869e-10	0.09	4.749e-12	0.04
8	2.668e-12	0.09	1.978e-14	0.05	3.645e-11	0.09	2.168e-13	0.05
9	2.558e-13	0.10	1.493e-15	0.08	3.592e-12	0.10	2.024e-14	0.09
10	2.520e-14	0.10	1.315e-15	0.88	3.704e-13	0.10	1.911e-14	0.94
11	2.930e-15	0.12	–	–	4.573e-14	0.12	–	–
12	1.356e-15	0.46	–	–	2.217e-14	0.48	–	–
C_Ω	**1.73**		**1.80**		**1.78**		**1.80**	
C_L	**1.88**		**2.41**		**2.01**		**2.42**	

	$M_\infty = 0.3$				$M_\infty = 0.46$			
grid size $\longrightarrow$	30×10		60×20		30×10		60×20	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	6.725e-02	-	5.072e-02	-	0.104e-00	-	7.933e-02	-
1	1.002e-02	0.14	6.992e-03	0.13	2.162e-02	0.21	1.321e-02	0.16
2	1.097e-03	0.11	5.183e-04	0.07	8.839e-03	0.41	5.217e-03	0.39
3	1.173e-04	0.11	5.436e-05	0.10	7.796e-03	0.88	5.063e-03	0.97
4	5.255e-06	0.04	1.650e-06	0.03	1.462e-03	0.19	1.030e-03	0.20
5	2.775e-07	0.05	1.107e-07	0.07	9.841e-04	0.67	3.262e-04	0.32
6	1.556e-08	0.06	2.566e-09	0.02	9.841e-05	0.10	3.372e-05	0.10
7	9.057e-10	0.06	2.278e-10	0.09	7.197e-06	0.07	3.813e-06	0.11
8	5.511e-11	0.06	5.227e-12	0.02	5.207e-07	0.07	1.663e-07	0.04
9	3.537e-12	0.06	4.342e-13	0.08	4.302e-08	0.08	1.083e-08	0.07
10	2.474e-13	0.07	6.287e-14	0.14	3.765e-09	0.09	7.768e-10	0.07
11	3.402e-14	0.14	6.740e-14	1.07	3.177e-10	0.08	6.514e-11	0.08
12	2.609e-14	0.77	-	-	2.592e-11	0.08	5.153e-12	0.08
C_Ω	**1.67**		**1.81**		**2.02**		**1.98**	
C_L	**1.94**		**2.46**		**2.46**		**3.23**	



(a)



(b)

Figure 67: A magnified view of the central area of the diffuser. Shown are the finest and the first coarse-level created by: a) grid size 120×40, b) grid size 80×50. Both cases correspond to $M_\infty = 0.46$.

12.9 Nozzle

The next example is flow through a nozzle. The rocket engine nozzle has three functions: to produce thrust, to conduct the exhaust gases back to the free-stream, and to set the mass flow rate through the engine. A diagram of the nozzle can be seen in Figure 68. The coordinates of the nozzle which are depicted in Table 40, were obtained by a security institute. The nozzle has a rectangular section, a flat bottom wall, and a converging-diverging channel with a maximum angle of 30° at the top wall. The convergent part follows a curved contour while the contour of the divergent part is a straight line. The ratio of the inlet area to the throat area is $\frac{A_{in}}{A_{throat}} = 1.4114$ and the ratio of the exit area and the throat area is $\frac{A_{out}}{A_{throat}} = 1.5$. Figure 69 is the mesh used for the calculation. The mesh size is 96×48 and it is clustered close to the upper and lower walls vertically and in the throat horizontally , with a stretching factor of 1.1.

Table 40: Coordinates of the Nozzle.

$x, [m]$	$y, [m]$	$x, [m]$	$y, [m]$
-0.20000	0	-0.03055	0.04721
0.06100	0	-0.02835	0.04520
-0.20000	0.11620	-0.02617	0.04330
-0.19000	0.11620	-0.02402	0.04139
-0.18078	0.11620	-0.02188	0.03949
-0.17225	0.11620	-0.01976	0.03761
-0.16433	0.11620	-0.01765	0.03574
-0.15694	0.11620	-0.01555	0.03389
-0.15003	0.11620	-0.01346	0.03208
-0.14355	0.11620	-0.01138	0.03022
-0.13746	0.11620	-0.00930	0.02840
-0.13170	0.11620	-0.00723	0.02663
-0.12626	0.11620	-0.00516	0.02479
-0.12111	0.11620	-0.00309	0.02311
-0.11621	0.11620	-0.00102	0.02167
-0.11154	0.11582	0.00106	0.02076
-0.10709	0.11420	0.00314	0.02030
-0.10283	0.11058	0.00523	0.02034
-0.09875	0.10718	0.00733	0.02083
-0.09484	0.10385	0.00944	0.02147
-0.09108	0.10063	0.01157	0.02211
-0.08745	0.09752	0.01371	0.02278
-0.08396	0.09450	0.01587	0.02345
-0.08059	0.09156	0.01805	0.02412
-0.07732	0.08872	0.02026	0.02476
-0.07416	0.08595	0.02249	0.02544
-0.07109	0.08325	0.02474	0.02608
-0.06811	0.08066	0.02703	0.02674
-0.06522	0.07812	0.02935	0.02743
-0.06239	0.07559	0.03170	0.02812
-0.05964	0.07316	0.03410	0.02882
-0.05695	0.07078	0.03653	0.02952
-0.05433	0.06844	0.03902	0.03022
-0.05175	0.06614	0.04155	0.03092
-0.04924	0.06389	0.04413	0.03163
-0.04676	0.06172	0.04677	0.03234
-0.04434	0.05956	0.04947	0.03305
-0.04195	0.05739	0.05224	0.03380
-0.03961	0.05527	0.05508	0.03455
-0.03730	0.05323	0.05800	0.03532
-0.03502	0.05118	0.06100	0.03610
-0.03277	0.04915	–	–

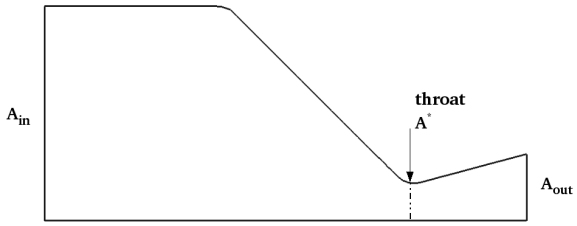


Figure 68: Nozzle diagram.

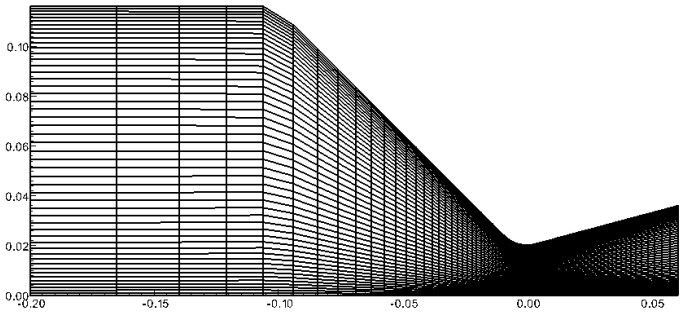
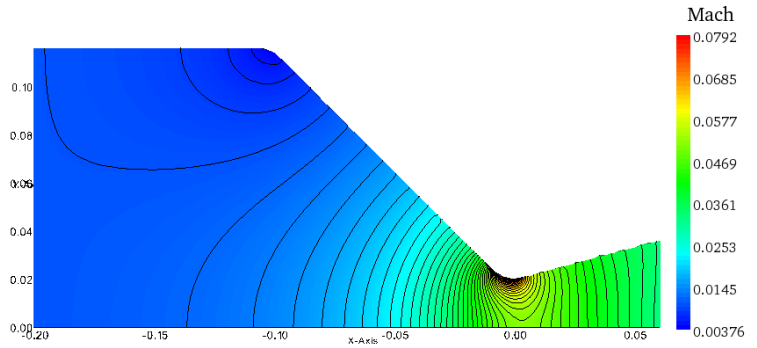


Figure 69: Mesh used for the converging-diverging nozzle flow test case.

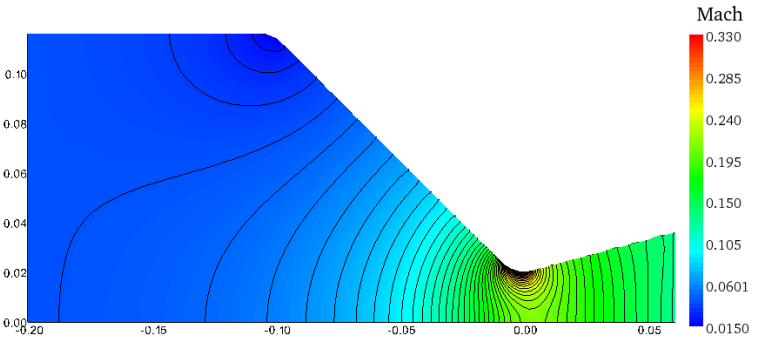
The following boundary conditions were specified:

- A subsonic flow at the inlet and the same mass flow rate at the outlet, in the x-direction.
- Solid-wall boundary condition at the top and bottom walls.

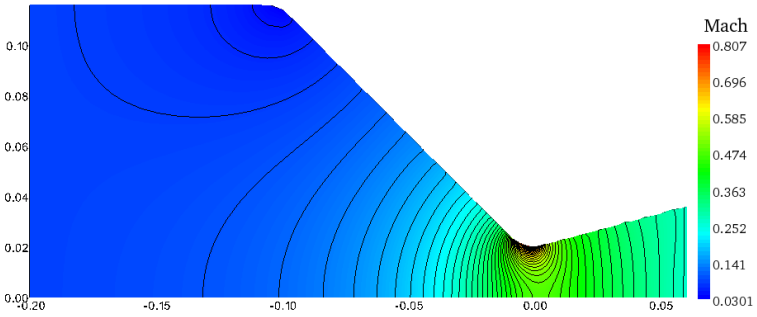
When the inlet velocity is not high enough to induce sonic flow in the throat, the flow in the nozzle is subsonic throughout. For a subsonic inlet velocity of $M_\infty = 0.092$ the area ratio exactly equals the critical ratio $^{A_{in}}/_{A^*}$ described in Eq. (204). The flow in the throat becomes sonic and a normal shock can be observed in the diverging section as can be seen in Figure 70. The pressure coefficient C_p computed along the nozzle upper wall for four cases of inlet velocity are presented in Figure 71. A strong pressure gradient is present in the case of $M_\infty = 0.092$, where a shock is obtained at the diverging section of the nozzle.



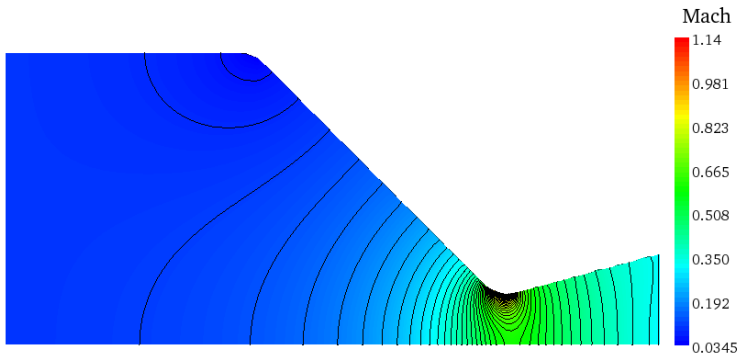
(a)



(b)



(c)



(d)

Figure 70: Distribution of velocity as computed on the (80×50) mesh, for the following velocities: a) $M_\infty = 0.01$, b) $M_\infty = 0.04$, c) $M_\infty = 0.08$, d) $M_\infty = 0.092$.

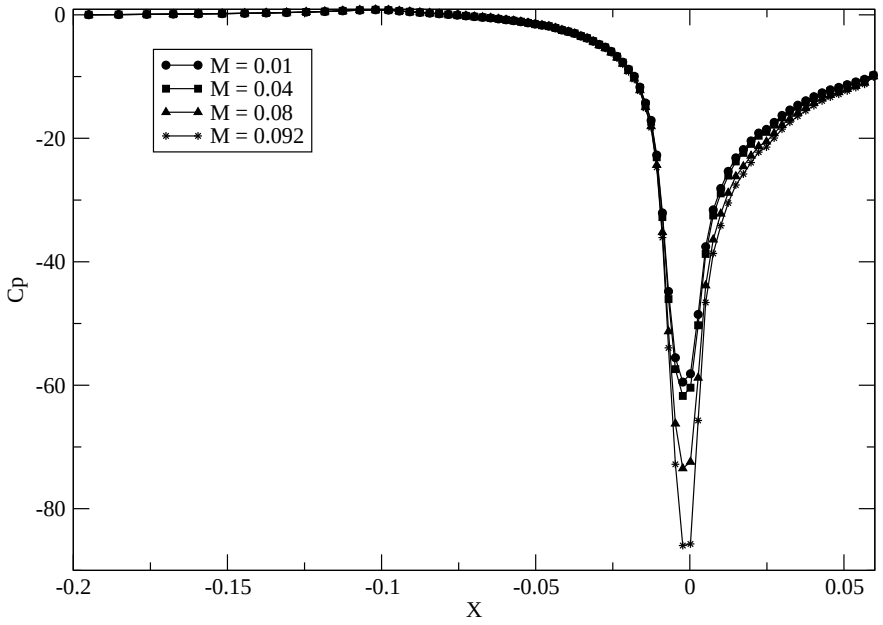


Figure 71: The pressure coefficient calculated on the upper wall for the above flow conditions.

12.9.1 AMG performance

The first coarse-level for each case described above is sketched in Figure 72. This coarsening pattern was obtained for all the Mach numbers, from $M_\infty = 0.01$ to $M_\infty = 0.092$. A dynamic threshold was applied for all the following cases since more stable performance were obtained in terms of convergence properties, mainly in the transonic case of $M_\infty = 0.092$. Several observations are in order here. First, this coarsening pattern is not so intuitive for this specific problem, since we would expect a uniform distribution of the coarse points at the inlet area, rather than a one-dimensional coarsening in the y-direction, at least for $M_\infty = 0.01$, where the equation is isotropic. A possible reason is the stretching of the grid cells in the x-direction, which contributes to the strong connections in the y-direction, and thus, the AMG coarsening algorithm automatically coarsens in the direction of the strong connections. As for the coarsening at the throat, it is what we would expect for all the free-stream Mach numbers, in particular for $M_\infty = 0.092$ where the anisotropy is largest.

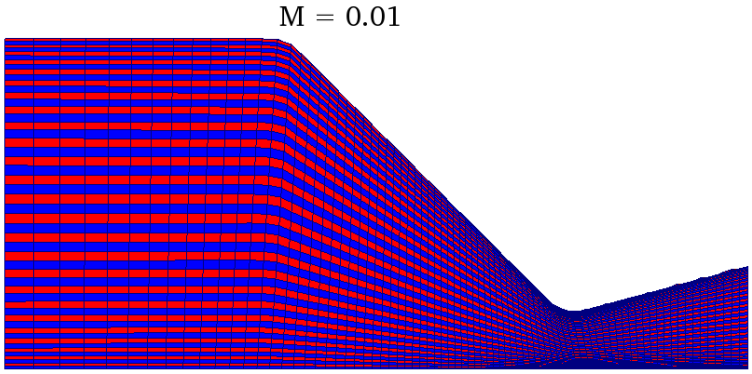


Figure 72: The finest and the first coarse-level for mesh size of (80×50). The red color corresponds to the C-point and the blue color corresponds to F-point. This coarsening pattern was obtained for $M_\infty = 0.01$. The same coarsening pattern was also obtained for various flow conditions as follows: $M_\infty = 0.04$, $M_\infty = 0.08$, and $M_\infty = 0.092$.

Table 41 presents the discrete second norm of the residual for each case of Mach number described above. Both the Mach contours and the convergence history are evidence that the AMG solves the problem to the level of discretization on each grid. The residual norm decreased rapidly for 10 to 12 V-cycles with the value in the corresponding $\|R^m\|_2$. We observe that each cycle exhibits a very stable convergence behavior with an asymptotic convergence rate of less than an order of magnitude. In the first case of $M_\infty = 0.01$, the setup phase was implemented only once, followed by 12 V-cycles. The case of $M_\infty = 0.04$, where the compressibility became significant, the results were obtained by repeating the setup phase four times while applying three V-cycles between each update. This results in a total of 12 V-cycles until the residual decreased to the desired level of $\tilde{}10^{-10}$. In the last two cases of $M_\infty = 0.08$ and $M_\infty = 0.092$, more frequent updates of the matrices A, $\tilde{A}$, and the restriction and interpolation operators were needed in order to achieve efficient and stable performance. In these cases, the setup phase was repeated six times while each update was followed by two V-cycles. It is important to note that the convergence rate in the first 2-3 cycles is not significantly damaged, compared to the previous problems, mainly due to a relatively good initial condition.

The convergence histories for the five cases are depicted in Figure 73. The effect of the Mach number on the AMG performance is clearly shown. The case of $M_\infty = 0.092$ results in a supersonic flow regime that is terminated by a shock wave. Convergence is somewhat slower here. It requires nearly twice the number of V-cycles (10) to decrease the L_2-norm of the residual to a level of 10^{-12}, compared to the case of $M_\infty = 0.01$. This slow convergence is mainly due to the $2-3$ first "waste" cycles (slow to converge due to strong nonlinearity) until the convergence is stabilized. Also it is very important to mention that, although there is strong anisotropy and nonlinearity, no extra local smoothing sweeps were needed at all to achieve convergence.

The grid complexity and operator complexity are presented in Figure 42. It is clear that the complexities are reasonable and bounded for all the flow velocities.

In the case of $M_\infty = 0.01$, the second pass process is applied for all the coarse-levels and it is well reflected in the grid complexity, which is slightly increased to $C_\Omega = 1.98$. Unfortunately, the operator complexity is not adversely affected. As for the remaining three cases, the second pass was applied only for the first coarse-level.

Table 41: Results of AMG V-cycles applied to the flow through a nozzle. The second norm of the residual $\|R^m\|_2$ and the convergence factor C_f are presented for each AMG V-cycle. The mesh size is 80×50.

	$M_\infty = 0.01$		$M_\infty = 0.04$		$M_\infty = 0.08$		$M_\infty = 0.092$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	1.019e-04	–	4.085e-04	–	8.207e-04	–	2.460e-03	–
1	5.200e-05	0.5	3.041e-04	0.74	4.176e-04	0.5	1.377e-03	0.55
2	1.389e-06	0.03	8.848e-06	0.03	5.672e-05	0.14	1.242e-04	0.09
3	4.495e-08	0.03	2.632e-07	0.03	2.271e-05	0.40	6.550e-05	0.53
4	1.536e-09	0.03	1.668e-08	0.06	1.279e-06	0.06	8.958e-06	0.14
5	6.144e-11	0.04	8.932e-10	0.05	9.215e-08	0.07	5.395e-06	0.60
6	2.511e-12	0.04	3.978e-11	0.04	2.977e-09	0.03	2.411e-07	0.04
7	1.220e-13	0.05	2.337e-12	0.06	7.656e-11	0.03	1.278e-08	0.05
8	6.721e-15	0.06	1.139e-13	0.05	4.182e-12	0.05	6.050e-10	0.05
9	4.062e-16	0.06	6.697e-15	0.06	1.793e-13	0.04	3.271e-11	0.05
10	6.957e-17	0.17	1.102e-15	0.16	8.592e-15	0.05	1.489e-12	0.05
11	6.475e-17	0.93	1.017e-15	0.92	5.929e-16	0.07	6.984e-14	0.05
12	6.555e-17	1.01	1.016e-15	1.00	4.248e-16	0.72	3.314e-15	0.05

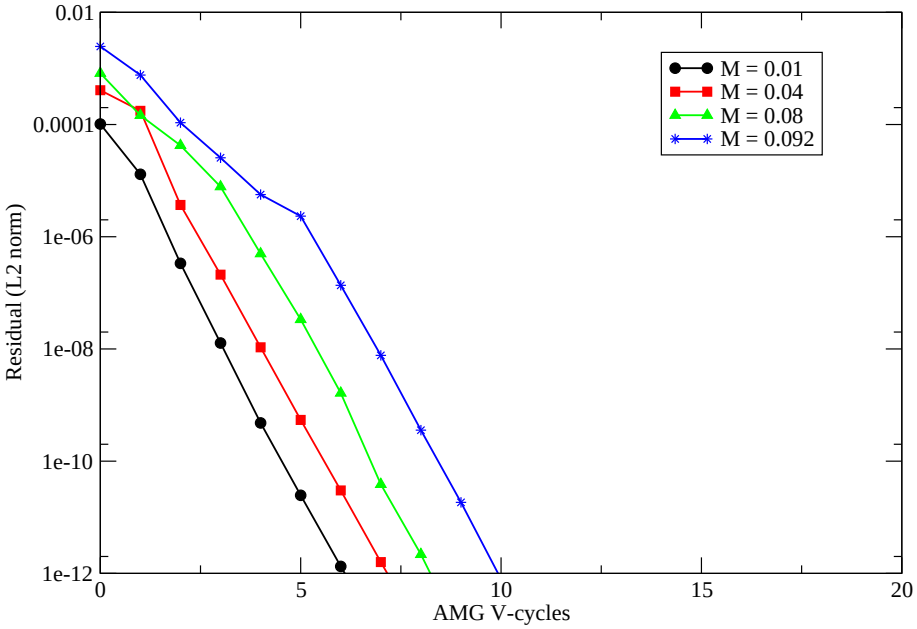


Figure 73: Convergence histories.

Complexities	$M_\infty = 0.01$	$M_\infty = 0.04$	$M_\infty = 0.08$	$M_\infty = 0.092$
C_Ω	1.98	1.92	1.90	1.93
C_L	2.47	2.47	2.52	2.46

Table 42: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

This problem was solved for two more cases of mesh sizes of 40×25 and 20×12 points. Both the L_2-norm of the residual and the convergence factor for two cases of $M_\infty = 0.01$ and $M_\infty = 0.092$ are presented in Table 41 . We deliberately decided to present a relatively simple case of subsonic flow compared to a more complex one characterized with a strong anisotropy. Observe that the difference of the above meshes hardly influences the convergence properties. As mentioned, computational work is determined by the operator complexity and the convergence factor. Only if both are bounded as a function of the problem size do we have a robust algorithm, namely, an asymptotically performance. In this case we can see that the complexities C_L and C_Ω are indeed independent of the grid size.

Let us examine the coarsening statistics in order to understand how AMG performs on this problem. Table 43 presents the number of rows and the number of the nonzeros for two cases of $M_\infty = 0.01$ and $M_\infty = 0.092$. The case of $M_\infty = 0.01$

was solved with seven levels (including the fine-level) while the case of $M_\infty = 0.092$ was solved with five levels only. In the latter case the coarsest level consists of a relatively large number of points (the reason was already pointed out in Section 8). The coarsening is not so fast and is similar for both cases. The first coarse-level is exactly half the number of points on the fine-level, while this reduction ratio of grid points for the rest of the coarse-levels is approximately reserved. A possible way to improve the coarsening performance is by using aggressive coarsening, but the cost is a much slower convergence rate mainly due to the interpolation process, which is significantly damaged. It is interesting to note here that although the stencils on the coarse-levels become larger, for both cases, subsequent coarsening then do not become faster. For both cases ($M_\infty = 0.092$ and $M_\infty = 0.01$) it happens simply because the dynamic threshold tends to produce large coarse-levels. In the case of $M_\infty = 0.01$, an additional effect is the second-pass process applied to all the coarse-levels. These two processes bound the coarsening speed but the gain has very efficient convergence properties.

Table 43: The results of the AMG V-cycles applied to the flow through a nozzle. The second norm of the residual after each V-cycle $\|R^m\|_2$ and the convergence factor C_f are detailed for two different mesh sizes.

	$M_\infty = 0.01$				$M_\infty = 0.092$			
grid size $\longrightarrow$	20×12		40×24		20×12		40×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	5.102e-04	–	1.318e-04	–	2.603e-03	–	1.578e-03	–
1	1.850e-04	0.36	9.100e-05	0.69	5.220e-04	0.20	1.127e-03	0.71
2	1.123e-05	0.06	1.077e-06	0.01	2.040e-04	0.39	1.980e-04	0.18
3	6.497e-07	0.06	6.191e-08	0.06	8.749e-05	0.43	1.094e-04	0.55
4	4.256e-08	0.07	5.122e-09	0.08	1.593e-05	0.18	5.334e-05	0.49
5	3.283e-09	0.08	4.727e-10	0.09	1.139e-06	0.07	1.973e-06	0.04
6	2.784e-10	0.08	4.541e-11	0.10	8.919e-08	0.08	1.540e-07	0.08
7	2.451e-11	0.09	4.441e-12	0.10	7.047e-09	0.08	1.318e-08	0.09
8	2.192e-12	0.09	4.385e-13	0.10	5.390e-10	0.08	3.809e-10	0.03
9	1.979e-13	0.09	4.352e-14	0.10	4.182e-11	0.08	2.390e-11	0.06
10	1.798e-14	0.09	4.331e-15	0.10	3.253e-12	0.08	1.684e-12	0.07
11	1.656e-15	0.09	4.350e-16	0.10	2.525e-13	0.08	1.213e-13	0.07
12	2.747e-16	0.17	1.052e-16	0.24	1.957e-14	0.08	8.929e-15	0.07
C_Ω	**1.90**		**1.91**		**1.75**		**1.85**	
C_L	**2.03**		**2.28**		**2.09**		**2.38**	

Table 44: Results of the AMG V-cycles applied to the flow through the nozzle in various Mach numbers.

	Number of rows		Number of nonzeros		Average entries per row	
Level	$M_\infty = 0.01$	$M_\infty = 0.092$	$M_\infty = 0.01$	$M_\infty = 0.092$	$M_\infty = 0.01$	$M_\infty = 0.092$
A^m	4000	4000	69424	69424	17.3	17.3
A^{m+1}	2000	2000	37023	44173	18.5	22
A^{m+2}	968	983	26865	27449	27.7	27.9
A^{m+3}	449	504	17971	19374	40	38.4
A^{m+4}	205	268	8541	10661	41.6	39.7
A^{m+5}	91	–	3219	–	35.3	–

12.10 Rocket engine

A rocket in its simplest form is a chamber enclosing a gas under pressure. A small opening at one end of the chamber allows the gas to escape, and it results in a thrust that propels the rocket in the opposite direction. The gas is produced by burning propellant that can be solid or liquid. A solid propellant is usually aligned along the inside walls of a cavity called the combustion chamber. The propellant is burned leading to build-up high temperature and pressure. The expanding gas escapes through the nozzle.

In this final numerical experiment I will analyze the flow through a solid propellant rocket, under all the assumptions of potential flow. The engine geometry (shown in Figure 74) is based on the nozzle from the previous example. The ratio of the chamber area to the throat area is $\frac{A_{in}}{A_{throat}} = 5.75$ and the ratio of the exit area and the throat area is $\frac{A_{out}}{A_{throat}} = 1.75$. The computational mesh is shown in Figure 75. The mesh size is 96×48 and it is clustered close to the upper and lower walls vertically and in the throat horizontally, with a stretch factor of 1.1.

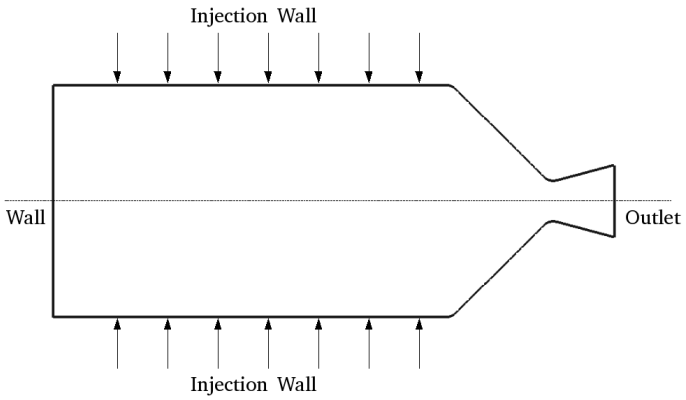


Figure 74: A diagram of the converging-diverging nozzle.

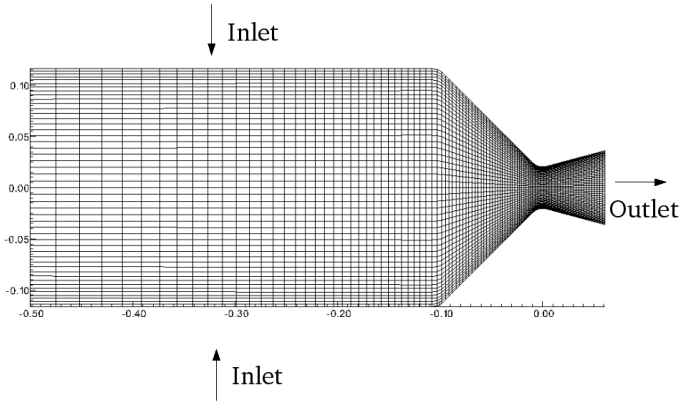


Figure 75: Mesh used for the 2D rocket test case.

12.10.1 Boundary conditions

1. The solid propellant is characterized by the bottom and upper walls (see the diagram). Turbulence effects significantly influence the flow processes close to the combustion surface. A full treatment of this region would include the modeling and resolution of complex physical and chemical phenomena that take place during the propellant combustion process. This process occurs in a very thin layer at the grain surface and is usually the subject of dedicated, detailed investigations, which are not part of this work. Therefore, in this simulation, the burning surface is simply treated as a wall through which mass injection is applied at a constant rate.

2. The mass conservation determines the outflow velocity through the nozzle. In practice, ghost cells were used in order to determine the axial velocity.

In order to perform an analysis of the flow in the rocket chamber with injection from the upper and lower walls, it is necessary to calculate the typical flow conditions (pressure, velocity, and sound speed) close to a burning propellant in a real life problem.

Before doing detailed quantitative analysis of the simulation results, the overall characteristics of flow field were examined to make sure that the solution reflected the expected behavior. All the flows that will be shown were computed on the 96×48 mesh. Figure 76 demonstrates the results of computations with subsonic and transonic flow. The Mach contours for the various cases of inlet flow velocities are considered in this investigation. The pressure coefficient distribution on the upper wall is plotted in Figure 77. One can readily show that when the Mach number near the upper and lower walls is $V_w = 0.028$ we get a choked throat and it is exactly the Mach number that results in the proper area ratio, as described in Eq. (204). The fifth case of $V_w = 0.03$ results in highly supersonic pockets with strong shock waves at the diverging section of the nozzle. The maximum Mach number at the shock was 2.28. It is clear that in this case the potential flow model is not valid anymore, since it implies a constant entropy and has therefore no mechanisms to generate entropy variations over the shock. The reason that this

case is demonstrated is that it is the closest case that describes a typical working point of a rocket engine. There is a very good agreement between these results and the theory as can be seen in Shapiro [14].

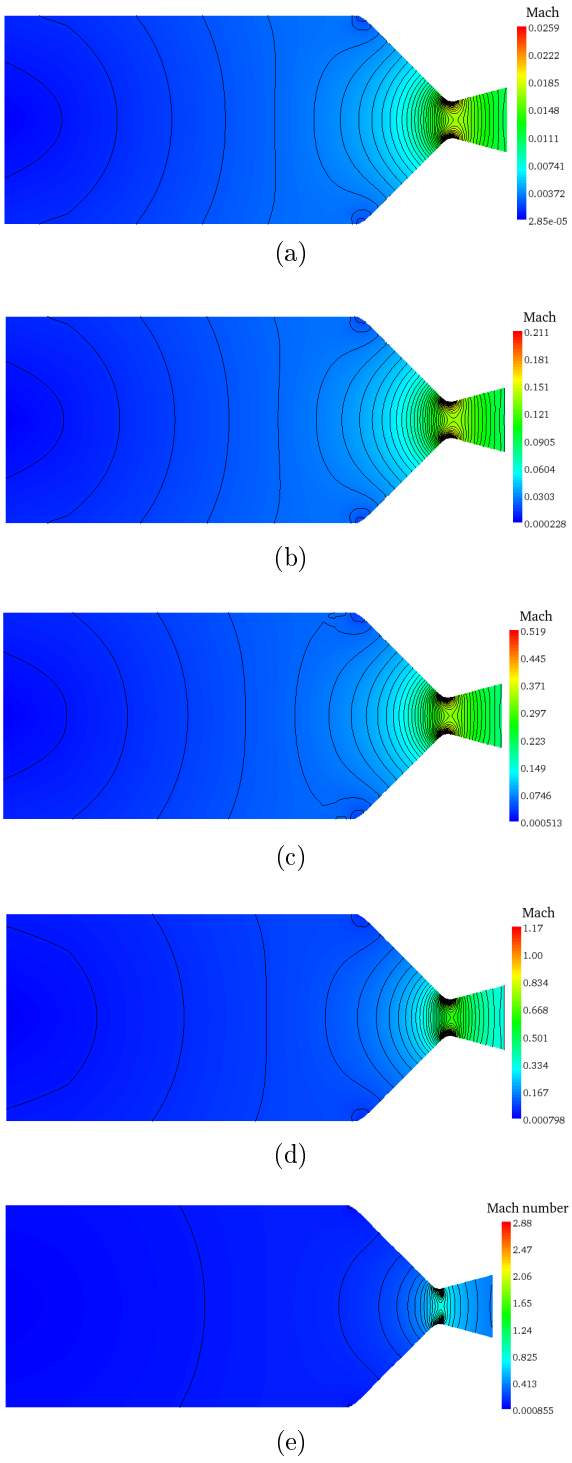


Figure 76: Mach number isolines computed on the (96×48) mesh, for the following flow conditions: a) $V_w = 0.001$, b) $V_w = 0.008$, c) $V_w = 0.018$, d) $V_w = 0.028$, e) $V_w = 0.03$. Observe the shocks in the diverging section appearing at $V_w = 0.028$ and $V_w = 0.03$.

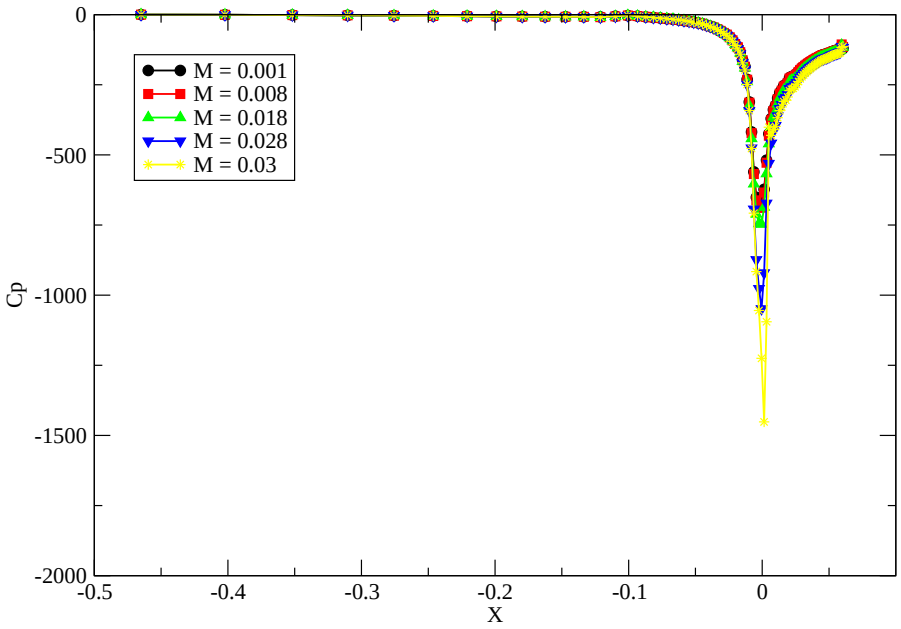
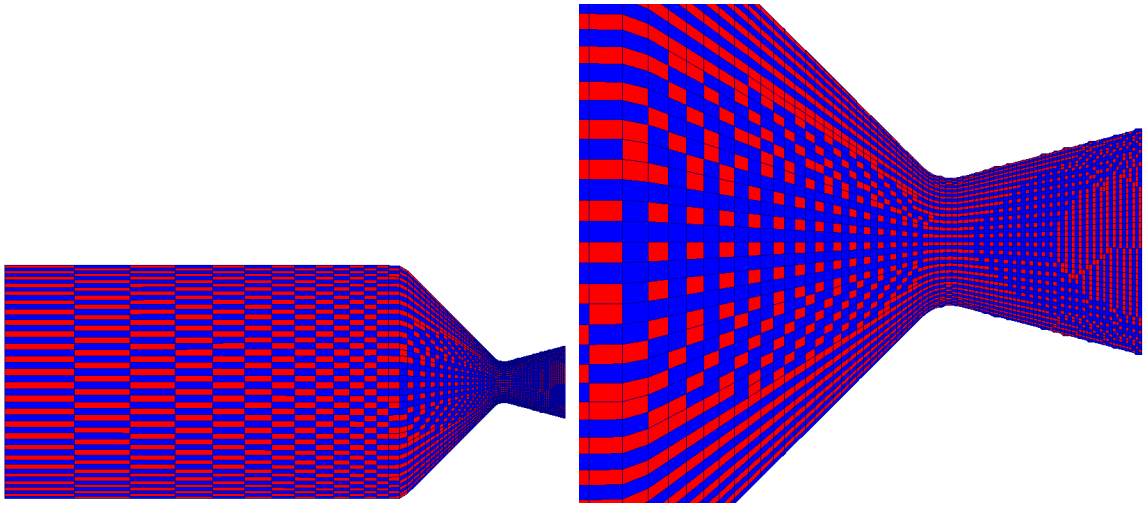


Figure 77: The distribution of pressure coefficient using a (96×48) mesh size for the following cases: $V_w = 0.001$, $V_w = 0.008$, $V_w = 0.018$, $V_w = 0.028$, and $V_w = 0.03$.

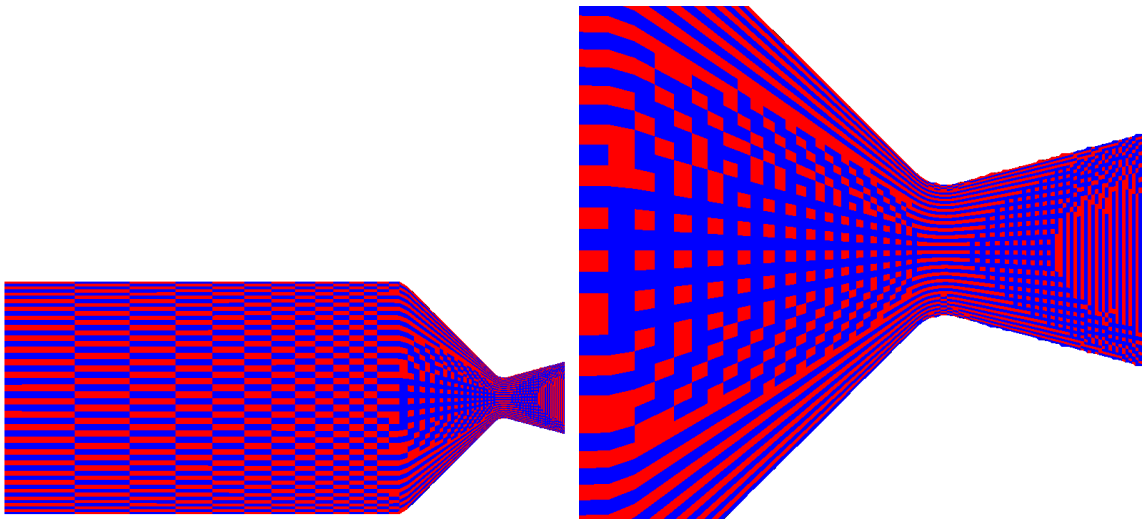
12.10.2 AMG performance

A fixed threshold parameter was imposed for the subsonic cases of $V_w = 0.001 - 0.018$, while for $V_w = 0.092$ the dynamic threshold was applied. The first coarse-level for each case described above is sketched in Figures 78 and 79. When the flow is incompressible, the operator is nearly isotropic and the points that construct the coarse-level are distributed uniformly in a red-black pattern, as can be seen in Figures 78(a). It is interesting to see that the coarsening in the throat area (where the distance between the upper and lower walls is minimal) is in the direction of the strong connectivity, in the y-direction, where the anisotropy is largest. The divergent and convergent areas are characterized by an increased velocity gradient. Due to the setting $\varepsilon = 0.25$, however, the coarsening algorithm still distributes the coarse points uniformly (possibly because it treats all the connections contained in this area as strong). Approaching the outlet region the cell's aspect ratio is increased and it contributes to strong connections in the x-direction. This effect is clearly visible in the one-dimensional coarsening pattern adjacent to the outlet area. As the Mach number increases more coarse points were chosen where the nonlinearity is largest (at the throat); however, the overall coarsening pattern is

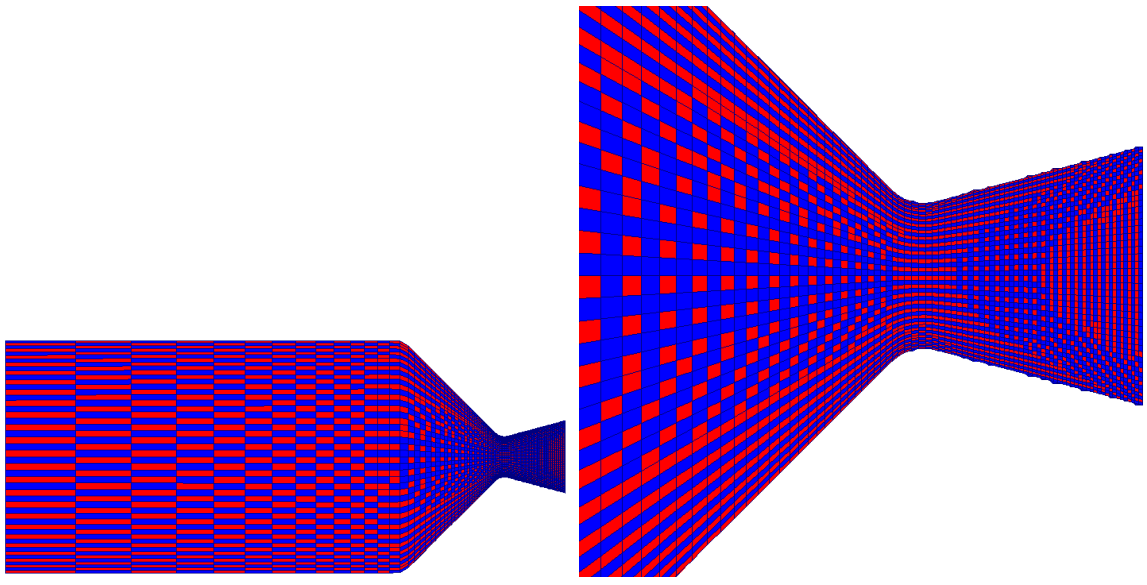
essentially the same. It is interesting that the situation was much different if a dynamic threshold was applied for this specific problem, as is sketched in Figure 80 for $V_w = 0.001$ and the same mesh size.



(a)

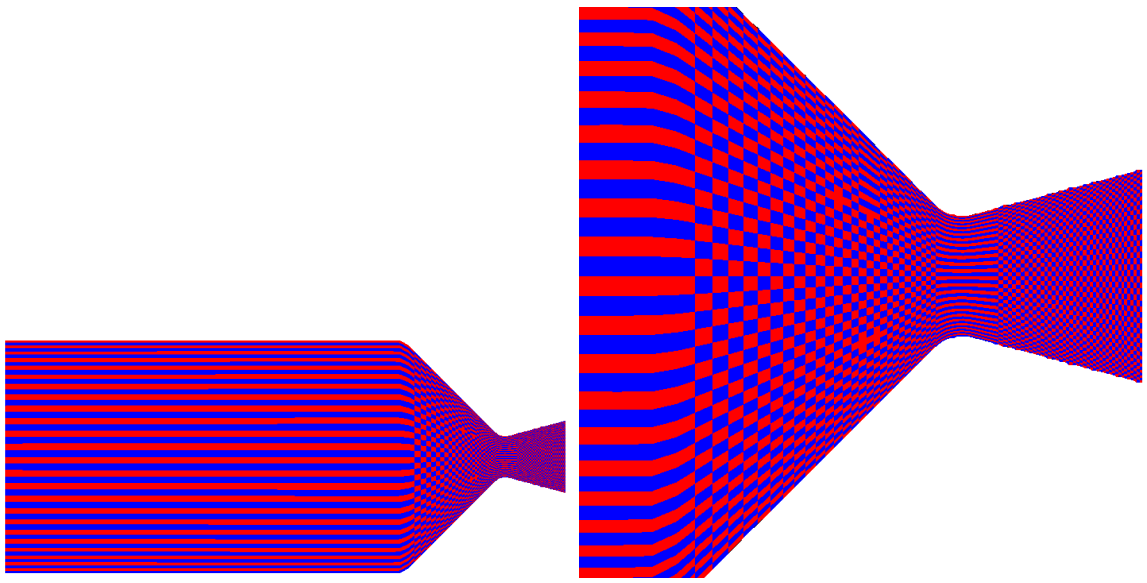


(b)

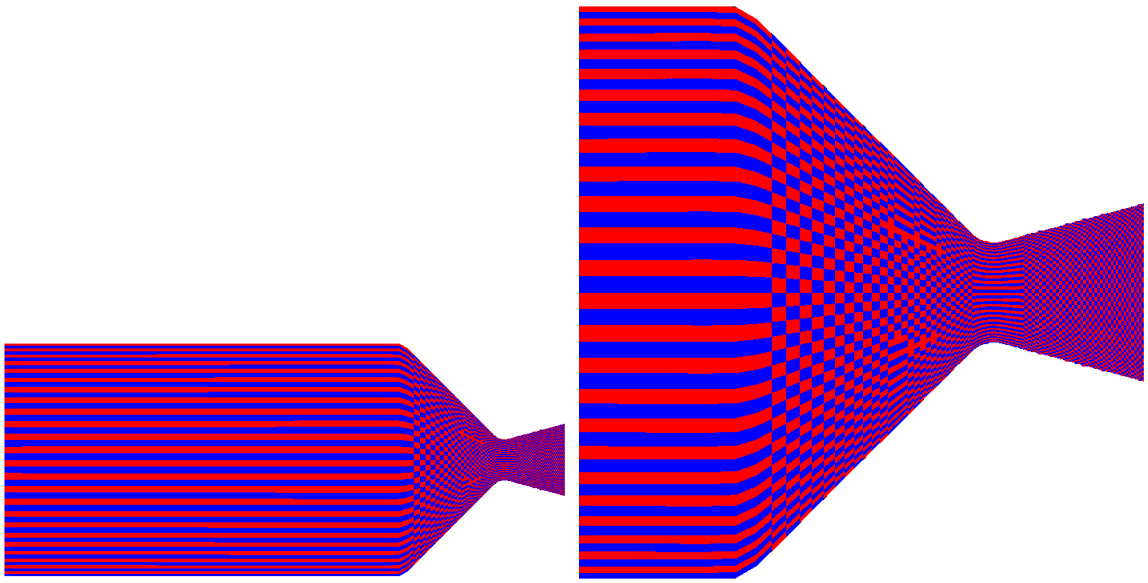


(c)

Figure 78: The finest and first coarse-level obtained for the flow through the rocket chamber for three Mach numbers: a) $V_w = 0.001$, b) $V_w = 0.008$, c) $V_w = 0.018$. Blue cell corresponds to an F-point and red cell corresponds to a C-point. The mesh size is (96×48). A fixed threshold was applied for all the these cases.



(a)



(b)

Figure 79: The finest and first coarse-level obtained for the flow through the rocket chamber for five Mach numbers: a) $V_w = 0.028$, b) $V_w = 0.03$. Blue cell corresponds to an F-point and red cell corresponds to a C-point. The mesh size is (96×48). A fixed threshold was applied for all the these cases.

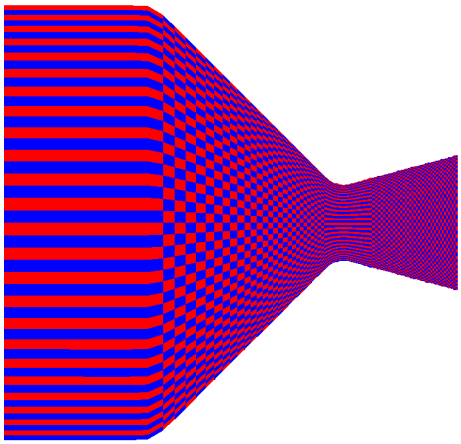


Figure 80: A magnified view of the rocket nozzle. The finest and first coarse-level obtained for the flow through the rocket chamber with $V_w = 0.001$. Blue cell corresponds to an F-point and red cell corresponds to a C-point. The mesh size is (96×48). A dynamic threshold was applied.

Convergence factor C_f and L_2-norm of the residual $\|R^m\|_2$ are presented in Table 45. We see that after 12 V-cycles, the residual L_2-norm reached 10^{-10} and the process attained an asymptotic convergence factor of 0.05 per V-cycle. Clearly, the AMG converges as rapidly for this problem as for the previous model problem (nozzle), where we saw a convergence factor of about 0.08. When $V_w = 0.001$ the equation is nearly isotropic and the entries in the coefficient matrices A, $\tilde{A}$, and A^* may vary, though also very little, even in the throat area where the velocity gradient is maximal. Consequently, the setup phase needs to be performed only once, at the beginning of the solving phase. This case was solved with 12 V-cycles. As the Mach number increases the effect of the compressibility becomes significant. In these cases the matrices including the transfer operators could not be "frozen" but rather updated while the setup phase is has to be performed more frequently. For instance, in the case of $V_w = 0.008$, the setup phase is repeated three times followed by four V-cycles each time. The cases of $V_w = 0.018$ and $V_w = 0.028$ required the setup phase to be performed six times with two V-cycles between each update.

In this context, the effect of the nonlinearity and the presence of shock waves is well observed in the convergence properties for $V_w = 0.028$, where the first three cycles are slow to converge and then the residual reduction is stabilized around 0.1 for the rest of the V-cycles. It takes around four "waste" cycles to converge close enough to the solution so that the nonlinearity effects stop hampering the further convergence process.

Figure 81 shows the convergence histories for the algorithm consisting of V-cycles, which was used for solving the problem for various Mach numbers. We can observe that the convergence behavior is comparable to that of the nozzle problem except that the convergence is slightly slower here. Ten cycles are needed in the fastest case, $V_w = 0.001$, to reduce the residual by 10 orders of magnitude.

Table 45: Results of the AMG V-cycles applied to the flow through a rocket chamber. The second norm of the residual $\|R^m\|_2$ and the convergence factor C_f are presented for each AMG V-cycle. The mesh size 96×48.

	$V_w = 0.001$		$V_w = 0.008$		$V_w = 0.018$		$V_w = 0.028$		$V_w = 0.03$	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	2.112e-04	–	4.741e-04	–	1.071e-03	–	1.533e-03	–	1.646e-03	–
1	2.467e-05	0.11	1.302e-04	0.27	2.217e-04	0.20	2.705e-04	0.17	2.951e-04	0.17
2	1.178e-06	0.05	8.108e-06	0.06	1.368e-05	0.06	1.343e-04	0.50	2.137e-04	0.72
3	5.791e-08	0.05	3.323e-07	0.04	7.179e-07	0.05	3.037e-05	0.23	1.363e-04	0.64
4	2.830e-09	0.05	1.650e-08	0.05	3.744e-08	0.05	1.051e-05	0.35	1.299e-04	0.95
5	1.399e-10	0.05	4.221e-10	0.03	2.947e-09	0.08	2.118e-06	0.20	7.072e-05	0.54
6	6.769e-12	0.05	3.154e-11	0.07	3.502e-10	0.12	5.294e-07	0.25	1.009e-05	0.14
7	3.372e-13	0.05	3.500e-12	0.11	4.283e-11	0.12	6.131e-08	0.12	2.629e-06	0.26
8	1.642e-14	0.05	4.799e-13	0.14	5.424e-12	0.13	9.002e-09	0.15	4.687e-07	0.18
9	8.320e-16	0.05	6.659e-14	0.14	6.841e-13	0.13	1.094e-09	0.12	7.186e-08	0.15
10	5.493e-17	0.07	9.483e-15	0.14	8.637e-14	0.13	1.558e-10	0.14	1.020e-08	0.14
11	3.527e-17	0.64	1.346e-15	0.14	1.089e-14	0.13	1.954e-11	0.14	1.412e-09	0.14
12	3.772e-17	1.07	2.389e-16	0.18	1.389e-15	0.13	2.680e-12	0.14	1.947e-10	0.14

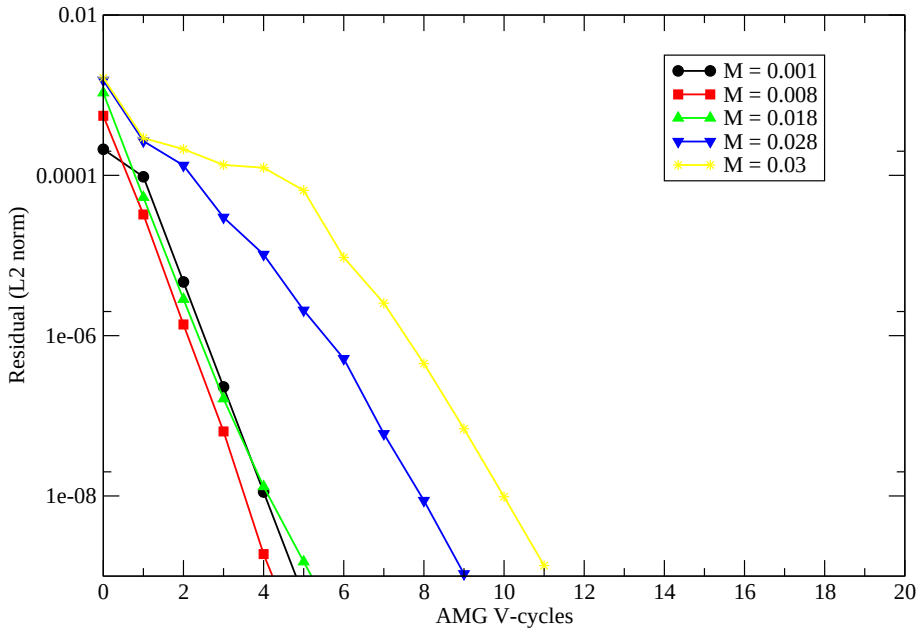


Figure 81: Convergence histories.

Table 46: Grid complexity C_Ω and operator complexity C_L for four cases of Mach number.

Complexities	$V_w = 0.001$	$V_w = 0.008$	$V_w = 0.018$	$V_w = 0.028$	$V_w = 0.03$
C_Ω	1.63	1.63	1.70	1.90	1.90
C_L	2.19	2.22	2.43	2.76	4.76

A grid convergence study has been conducted to verify the independence of the AMG algorithm on the grid resolution; three levels of resolution have been used with (32×12), (48×24), and (96×48) grid points in the axial and vertical directions. It is important to remember that our main interest here is to verify the robustness of the algorithm rather than achieving the greatest efficiency. Therefore in each resolution the AMG components (for instance, dynamic threshold, coarse-level parameter η, second pass) were "fixed", that is, they were not locally adjusted to particular requirements of a given resolution. These components remain exactly the same for all levels of resolution.

The L_2-norm of the residual $\|R^m\|_2$ and the convergence rate C_f are shown in Table 47. For each of the grid resolution the results demonstrate rapid convergence for all the four cases of Mach numbers. For each Mach number there is a minor degradation in the convergence rate with the increase of resolution. For instance, for the case $V_w = 0.008$ the convergence rate of $C_f = 0.1$ was obtained in mesh size (32×12) compared to $C_f = 0.15$ in mesh size (48×24). There are two possible reasons for this degradation. First, the interpolation is much less accurate as the resolution decreased, especially where the irregularity of the grid is dominant. Second, for small resolution the SGS method serves not only as a smoother (on the fine and coarse-levels as well) but also actually serves as a solver. In this case better convergence rates can be achieved.

Let us examine the storage requirements for four cases of Mach numbers. When the flow is subsonic, $V_w \leq 0.018$, the problem was solved with five levels and the properties are presented in Table 38. The coarsening is relatively fast while the initial coarse-level, A^{m+1}, has less than half the number of points on the finest level, A^m. Subsequent coarsening is even faster while each succeeding coarse-level has approximately one-third the number of points on the next finer level. The coarse-level operator on the second level can be seen to correspond to an average of a 30-point stencil. That is, although the reduction of points is substantial, the number of nonzeros in the second level matrix is still not far from that on the finest level. This fast coarsening (for $V_w \leq 0.018$) is well reflected in the relatively lower complexities presented in Table 46. For $V_w = 0.028$ the situation is different. The percentage of nonzeros is increased for the first coarse-level. Unfortunately, the operator complexity is slightly adversely affected (increase from $C_L \approx 2.2$ in the subsonic case to $C_L \approx 2.5$ in the transonic case). Considering the strong nonlinearity and extreme anisotropy of this problem, this increase of the complexities is quite acceptable.

Table 47: The results of AMG V-cycles applied to the flow through a rocket chamber. The second norm of the residual after each V-cycle $\|R^m\|_2$ and the convergence factor C_f are detailed for two different mesh sizes.

	$V_w = 0.001$				$V_w = 0.008$			
grid	24×12		48×24		24×12		48×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	3.555e-04	–	2.692e-04	–	2.846e-03	–	7.565e-04	–
1	5.183e-05	0.15	2.601e-05	0.09	4.549e-04	0.16	1.637e-04	0.21
2	3.660e-06	0.07	7.855e-07	0.03	3.451e-05	0.08	5.539e-06	0.03
3	2.444e-07	0.07	7.165e-08	0.09	3.021e-06	0.09	6.032e-07	0.11
4	1.858e-08	0.08	8.680e-09	0.12	3.032e-07	0.10	7.827e-08	0.13
5	1.506e-09	0.08	1.125e-09	0.13	3.176e-08	0.10	1.057e-08	0.14
6	1.242e-10	0.08	1.472e-10	0.13	3.339e-09	0.11	1.442e-09	0.14
7	1.029e-11	0.08	1.930e-11	0.13	3.491e-10	0.10	1.971e-10	0.14
8	8.545e-13	0.08	2.530e-12	0.13	3.626e-11	0.10	2.694e-11	0.14
9	7.095e-14	0.08	3.315e-13	0.13	3.749e-12	0.10	3.681e-12	0.14
10	5.873e-15	0.08	4.344e-14	0.13	3.864e-13	0.10	5.030e-13	0.14
11	4.998e-16	0.09	5.692e-15	0.13	3.966e-14	0.10	6.873e-14	0.14
12	6.390e-17	0.13	7.461e-16	0.13	4.190e-15	0.11	9.379e-15	0.14
C_Ω	**1.87**		**1.74**		**1.87**		**1.75**	
C_L	**2.17**		**2.28**		**2.17**		**2.28**	

	$V_w = 0.018$				$V_w = 0.028$			
grid	24×12		48×24		24×12		48×24	
V-cycle	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f	$\|R^m\|_2$	C_f
0	6.427e-03	–	4.873e-03	–	4.105e-03	–	2.672e-03	–
1	9.452e-04	0.14	4.840e-04	0.10	1.412e-03	0.34	5.101e-04	0.19
2	1.027e-04	0.11	3.435e-05	0.07	5.224e-04	0.37	2.697e-04	0.53
3	7.939e-06	0.08	2.673e-06	0.08	1.374e-04	0.26	5.996e-05	0.22
4	3.197e-07	0.04	9.496e-08	0.04	1.051e-05	0.08	1.023e-05	0.17
5	2.100e-08	0.07	3.444e-09	0.04	1.276e-06	0.12	2.583e-06	0.25
6	1.677e-09	0.08	1.148e-10	0.03	2.208e-07	0.17	4.325e-07	0.17
7	1.431e-10	0.09	5.066e-12	0.04	2.583e-08	0.12	6.744e-08	0.16
8	1.235e-11	0.09	2.110e-13	0.04	2.250e-10	0.01	1.020e-08	0.15
9	1.067e-12	0.09	1.028e-14	0.05	1.216e-11	0.05	1.523e-09	0.15
10	9.224e-14	0.09	5.259e-16	0.05	2.852e-13	0.02	2.260e-10	0.15
11	7.915e-15	0.09	1.201e-16	0.23	1.639e-14	0.06	3.342e-11	0.15
12	6.857e-16	0.09	1.162e-16	0.97	9.412e-16	0.06	4.937e-12	0.15
C_Ω	**1.82**		**1.79**		**1.79**		**1.66**	
C_L	**2.17**		**2.46**		**2.27**		**2.80**	

Table 48: Properties of the matrix $A^*$for AMG applied to the rocket engine problem. The mesh size is 96×48.

	Number of rows				Number of nonzeros				Average entries per row			
$V_w \longrightarrow$	0.001	0.008	0.018	0.028	0.001	0.008	0.018	0.028	0.001	0.008	0.018	0.028
Level												
A^m	4608	4608	4608	4608	80088	80088	80088	80184	17.3	17.3	17.3	17.4
A^{m+1}	2024	2011	2048	2307	57867	60373	62854	71928	28.6	30	30.7	31.1
A^{m+2}	648	650	829	1132	27398	27101	32176	39015	42.2	41.7	38.8	34.4
A^{m+3}	197	194	328	529	8857	8669	14326	20816	45	44.6	43.6	39.3
A^{m+4}	56	56	129	240	1836	1836	5735	9918	32.7	32.7	44.4	41.3

It was already mentioned that as the anisotropy is increased we expect a certain degradation of the AMG complexities since the final value of the operator complexity is influenced by the stencil size on the coarse-levels. As a consequence, the coarse-level operator tends to become larger towards coarser levels. However, this effect is limited by truncating the coarse-level matrices. The complexities presented above were obtained while truncating the coarse-level operator with a value of $\mu = 10^{-4}$. The L_2-norm of the residual, convergence rate, and the complexities are presented in Table 49, while applying $\mu = 10^{-5}$. Although the convergence is fast, the grid complexity is $C_\Omega = 2.01$, while the operator complexity substantially increased to $C_L = 5.32$. According to our requirements, this increase in the total solution cost is not acceptable.

Table 49: The results of AMG V-cycles applied to the flow through a rocket engine when the inlet Mach number is 0.028. The second norm of the residual after each V-cycle, the convergence factor, grid, and operator complexities are presented for 80×50 mesh size. The coarse-level truncation parameter is $\mu = 10^{-5}$.

	$V_w = 0.028$	
V-cycle	$\|R^m\|_2$	C_f
0	5.970e-03	–
1	3.334e-04	0.55
2	8.776e-05	0.26
3	2.364e-05	0.27
4	1.929e-06	0.08
5	8.074e-08	0.04
6	5.299e-09	0.07
7	2.955e-10	0.06
8	1.510e-11	0.05
9	7.448e-13	0.05
10	3.573e-14	0.05
11	1.737e-15	0.05
12	3.523e-16	0.20
C_Ω	**2.01**	
C_L	**5.32**	

13 Extending the 2D supersonic scheme to 3D

The 2D upwind numerical scheme for the supersonic flow regime can be extended to 3D discretization. Our main purpose is to demonstrate that such an extension ion will facilitate a construction of stable pointwaise relaxation, as well as in 2D. We are to do this while applying the same approach as applied in 2D (see Section 7). As already stated, the equation changes type when the flow becomes supersonic. This switching changes the diffusive character of the elliptic flow field to the propagation dominated behavior associated with the hyperbolic equation. Therefore, the discretization in the supersonic region must be inside the domain of dependence, within the characteristic cone. A common approach to implement this change is by using the rotated difference scheme that was introduced by Jameson [26]. In this part of the work, the difference scheme for the FPE in the quasi-linear form is designed to deal with problems characterized by flow in an arbitrary direction. The idea is to rearrange the equation as if it were locally expressed in a coordinate system aligned with the flow. The derivation of the three-dimensional FPE in the rotated difference scheme is based on the results introduced by Jameson [26]. First the 3D FPE in the quasi-linear form will be presented. More information on the derivation can be found in [12]. The steady continuity equation is given as follows:

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0. \tag{205}$$

Next, the x, y, and z momentum equations for an inviscid, steady fluid flow are written in a nonconservative differential form, after expanding and using the chain rule:

$$\begin{aligned}
u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} = -\frac{1}{\rho}\frac{\partial P}{\partial \rho}\frac{\partial \rho}{\partial x} = -\frac{a^2}{\rho}\frac{\partial \rho}{\partial x},\\
u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial \rho}\frac{\partial \rho}{\partial y} = -\frac{a^2}{\rho}\frac{\partial \rho}{\partial y},\\
u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial \rho}\frac{\partial \rho}{\partial z} = -\frac{a^2}{\rho}\frac{\partial \rho}{\partial z},
\end{aligned} \tag{206}$$

since from the isentropic relation of the pressure and density we get,

$$\frac{dP}{d\rho} = \frac{d\left[C\rho^{\gamma}\right]}{d\rho} = \gamma C\rho^{\gamma-1} = \gamma\frac{P}{\rho} = \gamma RT = a^2. \tag{207}$$

The potential function ϕ is now defined for 3D flow, $\frac{\partial \phi}{\partial x} = u$, $\frac{\partial \phi}{\partial y} = v$, $\frac{\partial \phi}{\partial z} = w$. Finally, the nonconservative FPE comes directly from substituting the terms of density $\frac{\partial \rho}{\partial x}$, $\frac{\partial \rho}{\partial y}$, and $\frac{\partial \rho}{\partial z}$ from (206) into the continuity equation,

$$\left(a^2 - u^2\right)\phi_{xx} + \left(a^2 - v^2\right)\phi_{yy} + \left(a^2 - w^2\right)\phi_{zz} - 2uv\phi_{xy} - 2uw\phi_{xz} - 2vw\phi_{yz} = 0. \tag{208}$$

Now, let us denote S as the stream direction, as sketched in Figure 82.

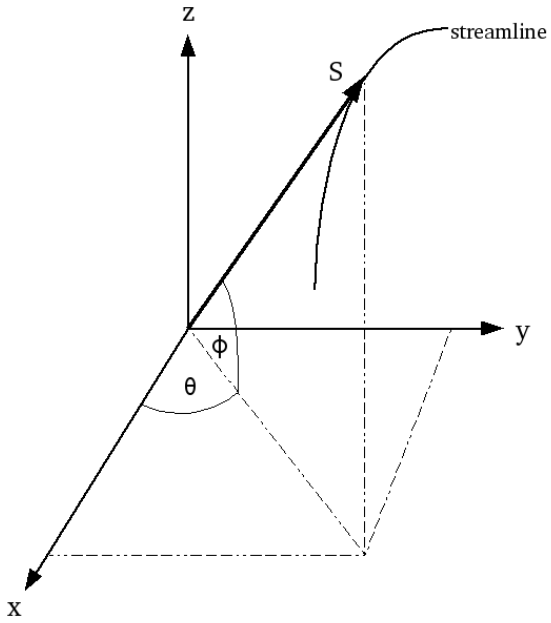


Figure 82: Cartesian coordinate system $s-n$ in 3D. The s-axis is aligned with the flow direction, and the n-axis is normal to it. The angle θ is the azimuth angle and the angle ψ is the zenith angle.

Then Eq. (208) can be written as follows:

$$\left(a^2 - q^2\right)\phi_{ss} + a^2\left(\nabla^2\phi - \phi_{ss}\right) = 0, \tag{209}$$

where q is the flow speed determined by $q^2 = u^2+v^2+w^2$ and $\nabla^2\phi = \phi_{xx}+\phi_{yy}+\phi_{zz}$. The direction of the flow is calculated as $\frac{u}{q} = cos\,(\theta)$, $\frac{v}{q} = sin\,(\theta)$, $\frac{w}{q} = cos\,(\psi)$, where θ is the azimuthal angle in the $x-y$ plane from the x-axis and ψ is the polar angle, measured from the $x-y$ plane to the velocity vector. The streamswise second derivative can be expressed as

$$\phi_{ss} = \frac{1}{q^2}\left(u^2\phi_{xx} + v^2\phi_{yy} + w^2\phi_{zz} + 2uv\phi_{xy} + 2vw\phi_{yz} + 2uw\phi_{xz}\right). \tag{210}$$

When the flow is subsonic all the derivatives are approximated by central difference formulas. If the flow is supersonic ($q > a$), then the terms appearing within ϕ_{ss} must be shifted in the direction of the flow while all contributions to the remaining terms are approximated by central difference formulas. The one-sided difference operators are biased in the upstream sense in all three coordinate directions. For simplification only a "narrow" approximation in the flow direction will be used. The discretized operators inside ϕ_{ss} are as follows:

$$
\begin{aligned}
\phi_{xx} &= \frac{(\phi_{i,j,k} - 2\phi_{i-1,j,k} + \phi_{i-2,j,k})}{\triangle x^2}, \\
\phi_{yy} &= \frac{\phi_{i,j,k} - 2\phi_{i,j-1,k} + \phi_{i,j-2,k}}{\triangle y^2}, \\
\phi_{zz} &= \frac{\phi_{i,j,k} - 2\phi_{i,j,k-1} + \phi_{i,j,k-2}}{\triangle z^2}, \\
\phi_{xy} &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k} - \phi_{i,j-1,k} + \phi_{i-1,j-1,k}}{\triangle x \triangle y}, \\
\phi_{xz} &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k} - \phi_{i,j,k-1} + \phi_{i-1,j,k-1}}{\triangle x \triangle z}, \\
\phi_{yz} &= \frac{\phi_{i,j,k} - \phi_{i,j-1,k} - \phi_{i,j,k-1} + \phi_{i,j-1,k-1}}{\triangle y \triangle z}.
\end{aligned}
\tag{211}
$$

All the remaining terms are approximated by central differencing:

$$
\begin{aligned}
\phi_{xx} &= \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\triangle x^2}, \\
\phi_{yy} &= \frac{\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}}{\triangle y^2}, \\
\phi_{zz} &= \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\triangle z^2}, \\
\phi_{xy} &= \frac{\phi_{i+1,j+1,k} - \phi_{i-1,j+1,k} - \phi_{i+1,j-1,k} + \phi_{i-1,j-1,k}}{4\triangle x \triangle y}, \\
\phi_{xz} &= \frac{\phi_{i+1,j,k+1} - \phi_{i-1,j,k+1} - \phi_{i+1,j,k-1} + \phi_{i-1,j,k-1}}{4\triangle x \triangle z}, \\
\phi_{yz} &= \frac{\phi_{i,j+1,k+1} - \phi_{i,j-1,k+1} - \phi_{i,j+1,k-1} + \phi_{i,j-1,k-1}}{4\triangle y \triangle z}.
\end{aligned}
\tag{212}
$$

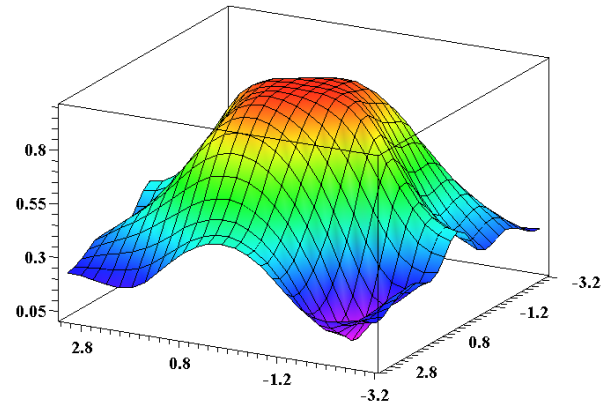
Since we restrict ourselves in this work to usage of a pointwise relaxation, while applying Gauss–Seidel and using Von-Neumann analysis, we get, as expected, an unstable scheme. Therefore the product operator $L\tilde{L}$ must be applied in order to achieve stability. From several numerical experiments we see that the difference operator resulting in the matrix $\tilde{A}$ (see section 8), for the cases of $0^o \leq \theta \leq 45^o$ was chosen to be the same as was applied in the 2D case but written in 3D form. Compared to the 2D case, there are many more possibilities to construct the operator $\widetilde{L}$, which results in a stable operator $L\tilde{L}$.

$$
\begin{aligned}
\tilde{L} = &\left(\frac{1}{4}sin^2(\theta) + \frac{1}{4}sin(\theta)cos(\theta)\right)\phi_{i-1,j+1,k} \\
&+ \left(\frac{1}{4} - \frac{1}{2}sin(\theta)cos(\theta)\right)\phi_{i+1,j+1,k} \\
&+ \frac{1}{2}sin^2(\theta)\phi_{i,j+1,k} \\
&+ \left(\frac{1}{2} - \frac{1}{2}sin^2(\theta)\right)\phi_{i+1,j,k} \\
&+ \left(\frac{1}{4}cos^2(\theta) + \frac{1}{4}sin(\theta)cos(\theta)\right)\phi_{i+1,j-1,k} - \phi_{i,j,k}.
\end{aligned}
\tag{213}
$$

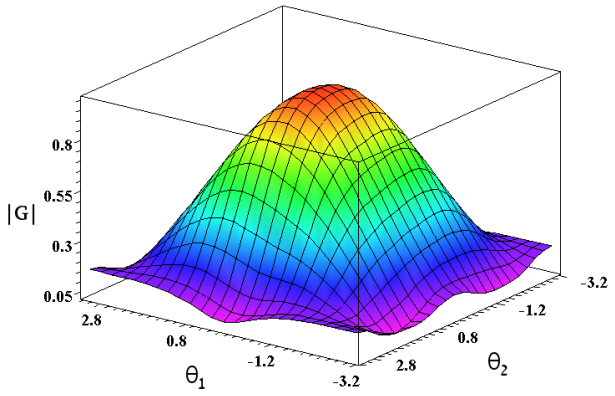
For simplicity the operator $\tilde{L}$ is discretized on the $x-y$ plane with no derivation in the z-direction. As a simple example, the product operator $L\widetilde{L}$ under the flow conditions $\theta = 0°$, $\phi = 0°$, and $M_\infty = 1.1$, is written as follows:

$$
\begin{aligned}
L\widetilde{L} &= 0.25\phi_{i+1,j-2,k,n} - 0.0525\phi_{i-1,j-1,k,n+1} - 0.23\phi_{i,j-1,k-1,n+1} - 0.79\phi_{i,j-1,k,n+1} \\
&+ 0.125\phi_{i,j-1,k+1,n} - 0.0525\phi_{i+1,j-1,k-2,n} + 0.21\phi_{i+1,j-1,k-1,n} - 0.21\phi_{i+1,j-1,k,n} \\
&+ 0.125\phi_{i+2,j-1,k-1,n} - 0.125\phi_{i+2,j-1,k+1,n} + 0.21\phi_{i-2,j,k,n+1} + 0.92\phi_{i-1,j,k-1,n+1} \\
&- 0.945\phi_{i-1,j,k,n+1} - 0.5\phi_{i-1,j,k+1,n} + 0.21\phi_{i,j,k-2,n+1} - 1.3\phi_{i,j,k-1,n+1} + 3.26\phi_{i,j,k,n+1} \\
&+ 0.25\phi_{i,j,k+1,n} - 0.105\phi_{i+1,j,k-2,n} - 0.08\phi_{i+1,j,k-1,n} - 0.92\phi_{i+1,j,k,n} + 0.5\phi_{i+1,j,k+1,n} \\
&+ 0.25\phi_{i+2,j,k-1,n} - 0.25\phi_{i+2,j,k+1,n} - 0.0525\phi_{i-1,j+1,k,n} - 0.23\phi_{i,j+1,k-1,n} - 0.79\phi_{i,j+1,k,n} \\
&+ 0.125\phi_{i,j+1,k+1,n} - 0.0525\phi_{i+1,j+1,k-2,n} + 0.21\phi_{i+1,j+1,k-1,n} - 0.21\phi_{i+1,j+1,k,n} \\
&+ 0.125\phi_{i+2,j+1,k-1,n} - 0.125\phi_{i+2,j+1,k+1,n} + 0.25\phi_{i+1,j+2,k,n}.
\end{aligned}
\tag{214}
$$

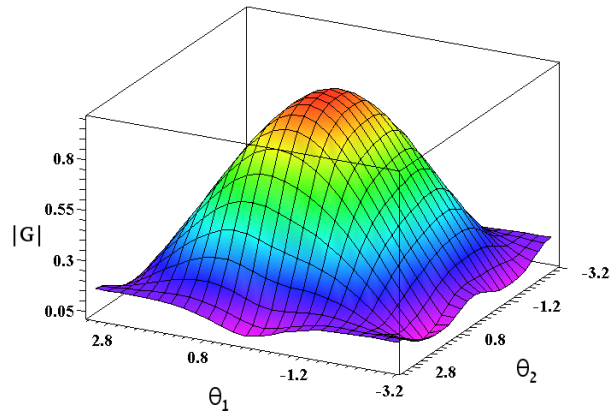
The numerical stability analysis of the product operator while applying the damped Gauss-Seidel relaxation, with a an under relaxation parameter of $\omega = 0.8$, was verified by Von-Neumann analysis. The stability condition requires that the modulus of the amplification factor should be lower than or equal to one. It is interesting to note that in the supersonic region, no additional terms were needed for the operator $\widetilde{L}$ written in 2D form, for the stability of the scheme, in all flow directions. Three views of the amplification factor as a surface over the regions $[-\pi, \pi] \times [-\pi, \pi] \times [0]$, $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi]$, and $[-\pi, \pi] \times [-\pi, \pi] \times [\pi]$, are given in Figures 87-86 for different flow conditions. The computations were done using the Maple mathematical commercial software (Maple version 11). If the operator is not stable, it is likely to be a result of the slower modes ($\theta_1 \sim 0$, $\theta_2 \sim 0$, $\theta_3 \sim 0$). Although the operator was checked for $-\pi \leq z \leq \pi$, for presentation three cases of $z = [-\pi, 0, \pi]$ are enough to present stability. Each surface corresponds to the variation of $|G(\theta_1, \theta_2, \theta_3)|$ over $-\pi \leq \theta_1 \leq \pi$ and $-\pi \leq \theta_2 \leq \pi$ for a fixed value of θ_3.



(a)

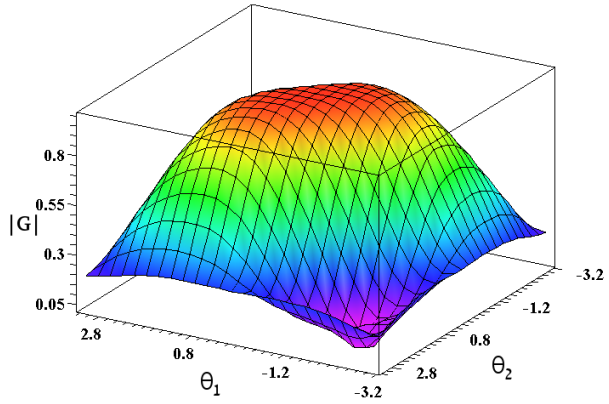


(b)

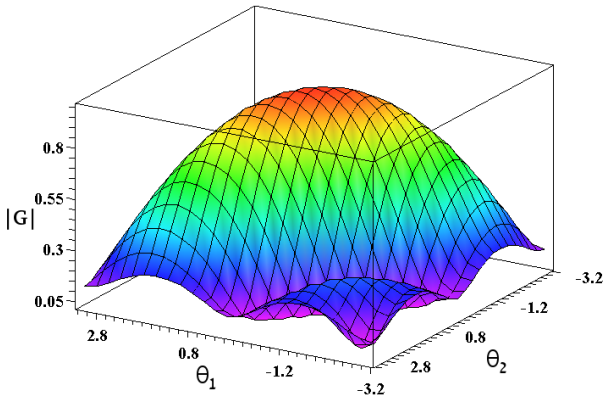


(c)

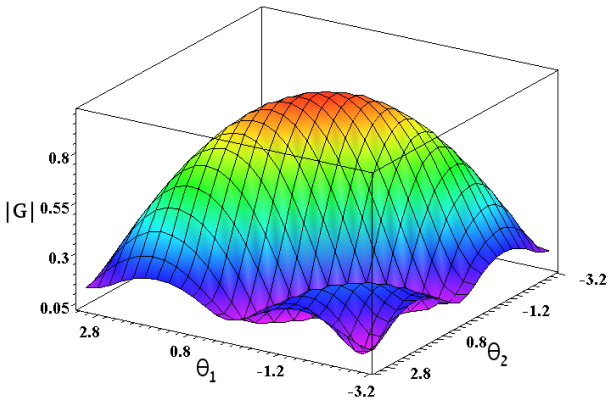
Figure 83: Amplification factor, $|G(\theta_1, \theta_2, \theta_3)|$, for the Gauss–Seidel method applied to the model problem in three dimensions, shown as a surface over the regions: a) $[-\pi, \pi] \times [-\pi, \pi] \times [0]$, b) $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi]$, c) $[-\pi, \pi] \times [-\pi, \pi] \times [\pi]$. The flow conditions are: $M_\infty = 1.1$, $\theta = 0°$, $\psi = 0°$.



(a)

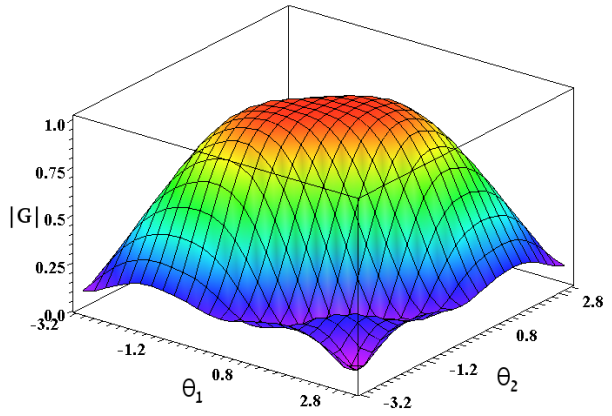


(b)

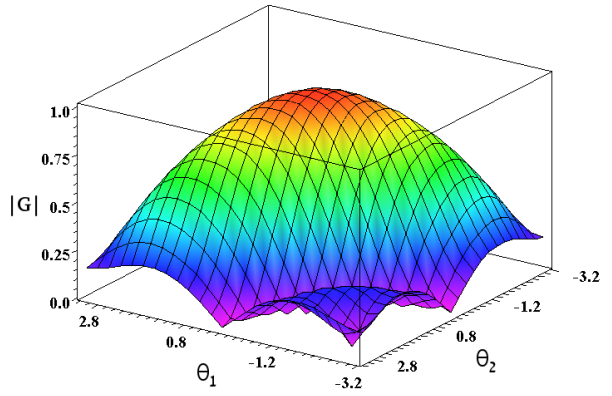


(c)

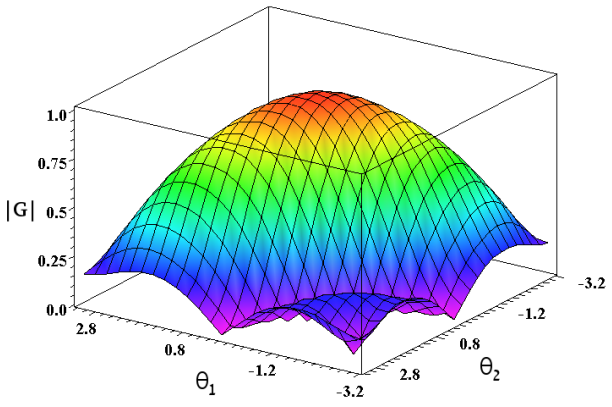
Figure 84: Amplification factor, $|G(\theta_1, \theta_2, \theta_3)|$, for the Gauss-Seidel method applied to the model problem in three dimensions, shown as a surface over the regions: a) $[-\pi, \pi] \times [-\pi, \pi] \times [0]$, b) $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi]$, c) $[-\pi, \pi] \times [-\pi, \pi] \times [\pi]$. The flow conditions are: $M_\infty = 1.1$, $\theta = 45°$, $\psi = 45°$.



(a)

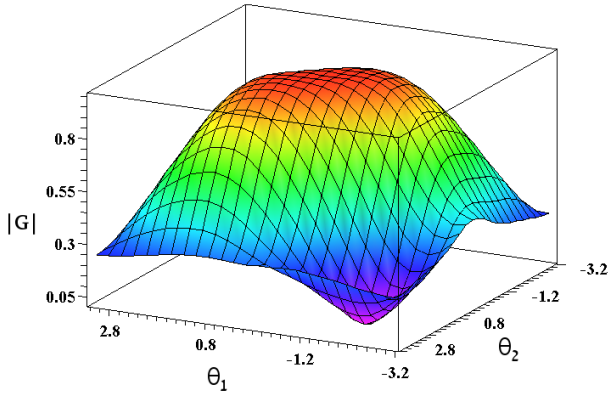


(b)

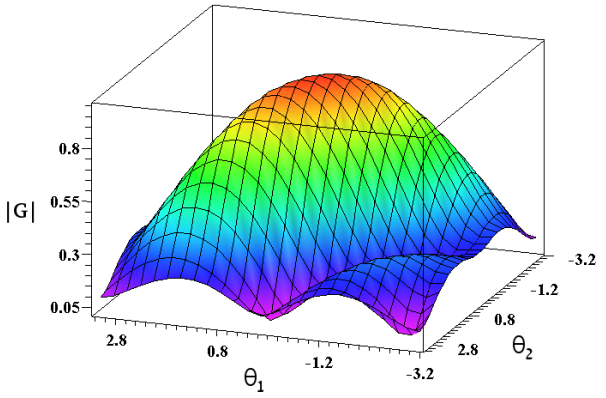


(c)

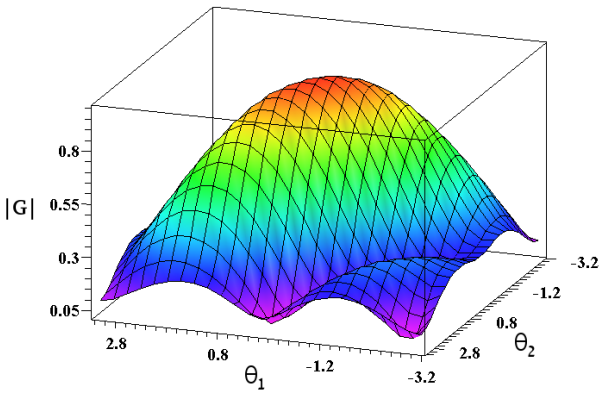
Figure 85: Amplification factor, $|G(\theta_1, \theta_2, \theta_3)|$, for the Gauss-Seidel method applied to the model problem in three dimensions, shown as a surface over the regions: a) $[-\pi, \pi] \times [-\pi, \pi] \times [0]$, b) $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi]$, c) $[-\pi, \pi] \times [-\pi, \pi] \times [\pi]$. The flow conditions are: $M_\infty = 1.1$, $\theta = 45°$, $\psi = 22.5°$.



(a)



(b)



(c)

Figure 86: Amplification factor, $|G(\theta_1, \theta_2, \theta_3)|$, for the Gauss-Seidel method applied to the model problem in three dimensions, shown as a surface over the regions: a) $[-\pi, \pi] \times [-\pi, \pi] \times [0]$, b) $[-\pi, \pi] \times [-\pi, \pi] \times [-\pi]$, c) $[-\pi, \pi] \times [-\pi, \pi] \times [\pi]$. The flow conditions are: $M_\infty = 1.1$, $\theta = 30$, $\psi = 60°$.

Some analysis and experimentation reveals that a better smoothing factor is obtained when the the operator $\widetilde{L}$ includes points in the z-direction. As an example, the discretization of $\widetilde{L}$ in the streamwise direction is as follows:

$$\widetilde{L} = \phi_{i+1,j,k} + \phi_{i+1,j+1,k} + \phi_{i+1,j-1,k} + \phi_{i+1,j,k+1} + \phi_{i+1,j,k-1} - \phi_{i,j,k}. \tag{215}$$

The amplification factor as a surface over the regions $[-\pi,\pi] \times [-\pi,\pi] \times [0]$, $[-\pi,\pi] \times [-\pi,\pi] \times [-\pi]$, and $[-\pi,\pi] \times [-\pi,\pi] \times [\pi]$, is given in Figure 87 for $M_\infty = 1.2$ and ($\theta = 0°$, $\psi = 0°$).

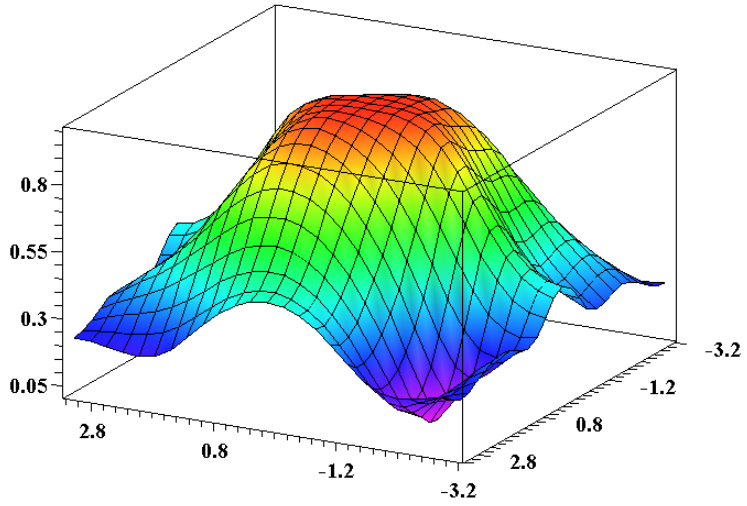


Figure 87: Amplification factor, $|G(\theta_1,\theta_2,\theta_3)|$, for the Gauss-Seidel method applied to the model problem in three dimensions, shown as a surface over the regions: a) $[-\pi,\pi]\times[-\pi,\pi]\times[0]$, b) $[-\pi,\pi]\times[-\pi,\pi]\times[-\pi]$, c) $[-\pi,\pi]\times[-\pi,\pi]\times[\pi]$. The flow conditions are: $M_\infty = 1.1$, $\theta = 30°$, $\psi = 60°$.

It is evident that the scheme is numerically stable whenever the velocity coincides with one of the three coordinate directions. These examples present good results, which indicate that the approach of solving the supersonic flow regime by applying a pointwise relaxation method is promising, both in the subsonic and supersonic flow regimes.

14 Conclusions

The objective of this thesis is to develop a highly efficient solver for the Full Potential Equation (FPE) that will be able to compute transonic external and internal flows attaining a (nearly) linear computational complexity. The FPE is obtained by the assumptions that the flow is inviscid and irrotational, and the Navier-Stokes equations are reduced down to a single equation. The FPE is useful for design and analysis of airfoil, wings, diffusers, etc. Computations are usually much less resource-consuming than those solving the Euler or Navier-Stokes equations. The FPE can be used for transonic flows, where a lot of design issues are of interest. The key innovation of this work is the solver's efficiency

and also the fact that it is achieved by means of adapting and applying the AMG approach to solving the problem.

The transonic flow problem is a rather complex one from the computational point of view. One of the main difficulties is the fact that the differential operator changes its type between elliptic for subsonic flow regime and hyperbolic (with respect to the flow direction) in the supersonic flow regime. Another (sub-)difficulty is that the subsonic flow regime itself presents two extremities (and all the possible cases between): nearly isotropic operator for the low speed case and a highly anisotropic operator for a nearly sonic flow speed. While the standard AMG algorithm can treat the latter with difficulty, it has never been applied yet, to the best of our knowledge, to the supersonic regime. The difficulties here begin with the fact that a simple pointwise relaxation procedure (a desirable component of AMG) appears unstable in the hyperbolic case. One of the main achievements of this work is the development of a pointwise relaxation procedure that is stable (and constitutes a good smoother - in the algebraic sense) for both the subsonic and supersonic flow regimes. Second, we constructed a variant of an AMG algorithm that employs the new relaxation procedure and allows to achieve very good convergence for both elliptic and hyperbolic cases.

We restricted ourselves in this work to use of a pointwise relaxation. Therefore, we have to ensure there is a variant of such a relaxation at our disposal that not only is stable for all the cases of interest, but also provides a good smoothing. While the simple damped Jacobi and Gauss–Seidel relaxation schemes are suitable for the subsonic case, both of them are unstable in the supersonic case. This was verified by the Von-Neumann analysis. In this work we developed a pointwise relaxation procedure that is stable (and constitutes a good smoother – in the algebraic sense) both in the subsonic and supersonic flow regimes. Second, we constructed a variant of an AMG algorithm that employs the new relaxation procedure and allows achieving a very good convergence in both elliptic and hyperbolic cases. This is demonstrated by a variety of numerical experiments concerning various flow regimes and flow directions (with respect to the grid). The results clearly show that the convergence rates and complexities are independent of the problem size (resolution).

The innovative aspects of the proposed AMG algorithm are as follows:

- The main achievement of this work is in developing a stable pointwise direction-independent relaxation for the supersonic and subsonic flow regimes. This development is a prerequisite for considering application of AMG to the transonic flow problem.

- An improved coarsening process is proposed. Instead of using a fixed threshold parameter in order to select the coarse-level points, we developed a dynamic threshold parameter as a measure of the strength of connection between the matrix variables. The coarsening by the dynamic threshold was shown to be less effective for certain elliptic problems (subsonic flow). However, for transonic and supersonic flow regimes, where the operator does not form an M-matrix, we obtained much better performance. In some cases where an irregular grid is involved, shock waves, and extreme nonlinearity, the dynamic threshold is very essential just in order to reach convergence.

- A new algorithm based on [93] was implemented for the construction of the restriction operator and the coarse-level equations. The proposed algorithm significantly improves the convergence rate for all the cases relevant for the purpose of this work.

- A modified formulation of the interpolation operator is presented. While the standard interpolation is suitable mainly for the M-matrix problems (subsonic flow), the proposed formula is more accurate and can be used for more general matrix problems. The proposed interpolation operator includes the choice of negative weights, which is necessary in some cases.

- Development of the FMG approach in the context of AMG - the MG V-cycles were replaced by AMG V-cycles. This approach significantly improved the initial condition, especially in nonlinear cases where the initial approximation is critical, to start the iterative procedure from a good initial approximation. This approach is very essential in our specific problems, for instance where a circulation calculation is involved.

In this research work we developed a structured 2D body-fitted grid FPE solver that is based on the AMG method. The flow solver is capable of resolving flows velocity from subsonic to transonic and supersonic regimes. The flow solver can accommodate complex geometries in different resolutions. The numerical performance is almost independent of the problem size. With these features, the present flow solver is a valuable tool for engineers. Solutions from the FPE are useful for design analysis and for initialization of Euler flow solvers. The efficiency of the constructed solver can be very useful during the design process where multiple computations need to be performed as small changes to the geometry are performed.

As a first problem to check how the AMG can deal with nonlinear problems (also in finite differences approach), we presented the capabilities of the algebraic multigrid to deal with the nonlinear small disturbance equation. The mathematical difficulties of the problem are associated primarily with the mixed hyperbolic and elliptic type of the equations and the presence of discontinuities. The computational method predicted the shock formation, and the problem was solved with a convergence factor of less than an order of magnitude.

Several two-dimensional flow calculations have been performed to test the performance of the algebraic multigrid method implemented on the FPE under the body-fitted structured grid configurations. The tests were chosen from two major aspects: First, the flow model has to agree with the potential flow limitations. Second, the idea is to check the capability of the algorithm to deal with irregular structured grids together with an equation that becomes extremely anisotropic near the sonic case and changes type to hyperbolic in the supersonic flow regime. Other critical areas of the algorithm were also discussed, including the Kutta-Joukowski condition and circulation. The Kutta-Joukowski condition is implemented to allow for computing flow around lifting airfoils. In order to evaluate the circulation efficiently we applied the FMG method in the context of AMG. As far as we know this is a new approach.

Finally, it was demonstrated that the approach used to construct 2D upwind numerical scheme for the supersonic flow regime and a stable pointwise relaxation

can be extended to 3D case. Since the discretization process results in a matrix A with a "stronger" diagonal, the 3D approach appears even more robust than the 2D one. This was verified by numerical stability analysis in various flow directions.

This algorithm is far from being optimal. The purpose of this work was not to design an algorithm which is fine-tuned for a specific problem with the aim to obtain the highest possible efficiency, but a robust algorithm which can address most of the relevant cases regarding the FPE and achieve a good performance without tuning its parameters any further. The AMG algorithm and all its components are constructed on the basis of compromise between the general applicability for the FPE and overall efficiency. Our main interest was to use the new algorithmic components and implement them as fixed strategies, rather than adjusting the algorithm to the particular requirement of a given specific problem.

References

[1] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91:99–165, 1963.

[2] J. W. Deardorff. A Numerical Study of Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers. *Journal of Fluid Mechanics*, 41:453–480, 1970.

[3] D. R. Chapman. Computational aerodynamics, development and outlook. *AIAA Journal*, 17:1293–1313, 1979.

[4] H. F. Joel and P. Milovan. *Computational Methods for Fluid Dynamics*, volume 3. Springer, 2002.

[5] S. C. Purohit, J. G. Shang, and W. L. Hankey. Numerical Simulation of Flow Around a Three-Dimensional Turret. *AIAA Journal*, 1020:1020, 1982.

[6] H. Rieger and A. Jameson. Solution of the Three-Dimensional Compressible Euler and Navier-Stokes Equations by an Implicite LU Scheme. *AIAA*, 88:619, 1988.

[7] H. Schock, D. J. Sosoka, and J. I. Ramos. Formation and Destruction of Vortices in a Motored Four-Stroke Piston Cylinder Configuration. *AIAA*, 22:948–949, 1984.

[8] T. Cebeci and A. M. O. Smith. Analysis of Turbulent Boundary Layers, New York. Academic Press., 1974.

[9] R. W. J. Newsome. Numerical Simulation of Near-Critical and Unsteady Sub-Critical Inlet Flow Fields. *AIAA*, 83:175, 1983.

[10] C. Hah. A Navier-Stokes Analysis of Three-Dimensional Turbulent Flow Inside Turbine Blade Rows at Design and Off-Design Conditions. *Trans. ASME, Journal of Engineering for Gas Turbines and Power*, 106:421–429, 1984.

[11] C. Fletcher. *Computational Techniques for Fluid Dynamics*, volume 2. Springer-Verlag, 1991.

[12] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1. John Wiley & Sons, 1988.

[13] T. H. Pulliam and J. L. Steger. Recent Improvements in Efficiency, Accuracy and Convergence for Implicite Approximate Factorization Algorithms. In *AIAA 23rd Aerospace Sciences Meeting*, volume 0360, Reno, January 1985.

[14] A. H. Shapiro. *The Dynamics and Thermodynamics of Compressible Fluid Flow*, volume 1. John Wiley & Sons, New York, 1953.

[15] J. D. Anderson. *Modern Compressible Flow: With Historical Perspective*. McGraw-Hill series in aeronautical and aerospace engineering, 2003.

[16] J. D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill Science Engineering, 2005.

[17] J. B. John and M. C. Russel. *Aerodynamics for Engineers*, volume 2. Prentice-Hall, 1989.

[18] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[19] A. S. Michel. *Compressible Fluid Flow*, volume 2. Prentice-Hall, 1993.

[20] E.M. Murman and J.D. Cole. Calculation of plane steady transonic flows. *AIAA*, 9(1):114–121, Jan 1971.

[21] E. M. Murman. Analysis of Embedded Shock Waves Calculated by Relaxation Methods. In *Proceeding First AIAA Computational Fluid Dynamics Conference*, Palm Springs, July 1973.

[22] J. L. Steger and H. Lomax. Transonic Flow About Two-Dimensional Airfoils by Relaxations Procedures. *AIAA*, 10:49–54, 1972.

[23] P. R. Garabedian and D. Korn. Analysis of Transonic Airfoils. *Communications on Pure and Applied Mathematics*, 24:841–851, 1971.

[24] W. F. Ballhaus and F. R. Bailey. Numerical Calculation of Transonic Flow About Swept Wings. *AIAA*, 72:677, 1972.

[25] F. R. Bailey and J. L. Steger. Relaxation Techniques for Three-Dimensional Transonic Flow About Wings. *AIAA*, 72:189, 1972.

[26] A. Jameson. Numerical Calculations of the Three-Dimensional Transonic Flow Over Yawed Wing. In *AIAA Computational Fluid Dynamics Conference*, pages 18–26, Palm Springs, July 1973.

[27] A. Jameson. Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1. *Pure. Appl. Math*, 27:283–309, 1974.

[28] A. Jameson. Transonic Potential Flow Calculations Using Conservative Form. In *Proceedings AIAA Second Computational Fluid Dynamics Conference*, Hartford, 1975.

[29] A. Jameson. Accelerated Iteration Schemes for Transonic Flow Calculations Using Fast Poisson Solvers. Technical report, ERDA Report, 1975.

[30] A. Jameson. Numerical Computation of Transonic Flows with Shock Waves. In Springer-Verlag, editor, *Symposium Transsonicum*, pages 384–414. Springer-Verlag, September 1975.

[31] A. Jameson and D. A. Caughey. A Finite Volume Method For Transonic Potential Flow Calculations . In *Proceedings AIAA Third Computational Fluid Dynamics Conference*, Albuquerque, 1977.

[32] A. Jameson and D. A. Caughey. Accelerated Iterative Calculations of Transonic Nacelle Flow Fields. *AIAA*, 15:1474–1480, 1977.

[33] A. Jameson and D. A. Caughey. Numerical Calculation of the Transonic Flow Past a Swept Wing. Technical Report C00-3077-140, ERDA, 1977.

[34] M. M. Hafez, J. C. South, and E. M. Murman. Artificial Compressibility Methods for the Numerical Solution of the Full Potential Equation. *AIAA*, 17:838–844, 1978.

[35] W. F. Ballhaus, A. Jameson, and J. Albert. Implicite Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems. *AIAA*, 16:573–579, 1978.

[36] T. J. Baker. Potential Flow Calculation by the Approximate Factorization Method. *AIAA*, 42:1–19, 1981.

[37] T. L. Holst. Fast, Conservative Schemes for the Full Potential Equation Applied to Transonic Flows. *AIAA*, 17:146–152, Feb. 1979.

[38] J. C. South and A. Brandt. Application of a Multi-Level Grid Method to Transonic Flow Calculations. In Jr. T. C. Adamson and M. F. Platzer, editors, *Transonic Flow Problems in Turbo-machinery*, pages 180–207. T. C. Adamson and M. F. Platzer, 1977.

[39] T. Holst. Implicite Algorithm for the Conservative Transonic Full Potential Equation Using an Arbitrary Mesh. *AIAA*, 17(10):1038–1045, 1979.

[40] A. Jameson. Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method. *AIAA*, 79:1458, 1979.

[41] A. Jameson. Remarks on the Calculation of Transonic Potential Flow by a Finite Volume Method. In Academic Press B. Hunt, editor, *Proc. IMA Conference on Numerical Methods in Fluids Dynamics*, pages 363–386. Conference on Numerical Methods in Fluids Dynamics, 1978.

[42] A. J. Van Der Wees, J. Van Der Vooren, and J. H. Meelker. Robust Calculation of 3D Transonic Potential Flow Based on the Non-Linear FAS Multi-Grid and Incomplete LU Decomposition. *AIAA*, 83:1950, 1983.

[43] T. L. Holst. Numerical Solution of the Full Potential Equation Using a Chimera Grid Approach. Technical Report TM 110360, NASA, 1995.

[44] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes. In *AIAA 14th Fluid and Plasma Dynamics*, number 81-1259, Reston, VA 1981. AIAA.

[45] E. Turkel and C. Swanson. A RK/Implicit Scheme for Solving the Compressibile Navier-Stokes Equations. In *ContactForum KVAB, Modern Techniques for Solving Partial Differential Equations*, June 2008.

[46] A. Brandt. Multigrid techniques: 1984 Guide with Applications to Fluid Dynamics. The Weizmann Institute of Science, Rehovot, Israel, 1984.

[47] I. Yavneh. *Multigrid Techniques for Incompressible Flows*. PhD thesis, The Weizmann Institute of Science, Rehovot, Israel, 1991.

[48] A. Brandt and I. Yavne. On Multigrid Solution of High Reynolds Incompressible Entering Flows. *J. Comput. Phys.*, 101(1):151–164, 1992.

[49] D. Sidilkover and E. J. Nielsen. Factorizable upwind schemes: the triangular unstructured grid formulation. In *15th AIAA CFD Conference*, number AIAA 01-2575 in Technical paper, Anaheim, California, June 2001.

[50] D. Sidilkover. A factorizable scheme for the equations of fluid flow. *Appl. Numer. Math.*, 41:423–426, 2002.

[51] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young. Parallel Newton–Krylov–Schwarz Algorithm for the Transonic Full Potential Equation. *SIAM J. Sci. Comput.*, 19:246–265, 1998.

[52] R.P. Fedorenko. A Relaxation Method for Solving Elliptic Difference Equations. *USSR Comput. Math. Math. Phys.*, 1:1092, 1962.

[53] R. P. Fedorenko. The Speed of Convergence of One Iterative Process. *USSR Comp. Math. Phys*, 4:227–235, 1964.

[54] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, Philadelphia, 2 edition, 2000.

[55] U. Trottenberg, A. Schuller, and C. W. Oosterlee. *Multigrid*. Elsevier Academic Press, 2001.

[56] S. F. McCormick, editor. *Multigrid Methods: Theory, Applications, and Supercomputing*, volume 110. Copper Mountain Conference on Multigrid Methods, Lecture Notes in pure and applied mathematics., 1987.

[57] S. Yair. *Matrix-Based Multigrid: Theory and Applications*, volume 2. Springer-Verlag, 2008.

[58] W. Hackbusch. *Multi-Grid Methods and Applications*. Number ISBN 3-540-12761-5. Springer Series in Computational Mathematics, 2003.

[59] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, New York, 1992.

[60] N.S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comput. Math. Math. Phys.*, 6:101, 1966.

[61] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems. In *in Proc. 3rd Int. Conf. on Numerical Methods in Fluids Mechanics*, pages 82–89. Springer-Verlag, 1973. Lecture Notes in Physics 18.

[62] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, 31:333–390, 1977.

[63] A. Brandt. Stages in developing multigrid solutions. In *Proceedings of the 2nd International Congress on Numerical Methods for Engineers*, 1980.

[64] W. Hackbusch and U. Trottenberg. Multi-Grid Methods. In *Proc. of the European Conference on Multigrid Methods*, Koln-Porz, Nov. 1981.

[65] W. Hackbusch and U. Trottenberg. Multigrid Methods II. In *Proc. of the European Conference on Multigrid Methods*, Cologne, Oct. 1985. European Conference on Multigrid Methods.

[66] W. Hackbusch and U. Trottenberg. Multigrid Methods III. In *Proc. of the European Conference on Multigrid Methods*, Bonn, Oct. 1990.

[67] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic Multigrid (AMG) for sparse matrix equations. In *Sparsity and Its Applications*. Cambridge University Press, Cambridge, 1984.

[68] A. Brandt. Algebraic multigrid theory: The symetric case. *Appl. Math. Comput.*, 19:23–56, 1986.

[69] K. Stueben. An introduction to Algebraic Multigrid. Technical report, German National Research Center for Information Technology (GMD), San Diego, 2001.

[70] J. W. Ruge and K. Stueben. Algebraic Multigrid. In *Multigrid Methods*, volume 3, pages 73–130. S. F. McCormick, ed., SIAM, Philadelphia, 1987.

[71] A. Brandt and B. Diskin. Multigrid solvers for the non-aligned sonic flow: The constant coefficient case. *Computers and Fluids*, 28:511–549, 1999.

[72] B. Diskin. *Efficient Multigrid Solvers for the Linearized Transonic Full Potential Equation*. PhD thesis, The Weizmann Institute os Science, 1998.

[73] A. Jameson. Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method. *Applied Mathematics and Computation*, 13:327–355, 1983.

[74] G. Volpe and A. Jameson. Transonic Potential Flow Calculations by Two Artificial Density Methods. In *4th Fluid Mechanics, Plasma Dynamics and Lasers Conference*, volume 86, 1986.

[75] R. D. Falgout. An introduction to Algebraic Multigrid. In *Computing in Science and Engineering*. Lawrence Livermore National Laboratory, 2006.

[76] Q. Chang, Y. Shu, and Wong H. Fu. On the Algebraic Multigrid. *Journal of Computational Physics*, 125:279–292, 1996.

[77] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.

[78] E. L. Wachspress. *Iterative Solution of Elliptic Systems*. Prentice Hall, 1966.

[79] D. Young. Iterative Solution of Large Linear Systems. Technical report, Academic Press, 1971.

[80] G. I. Marchuk. *Method of Numerical Mathematics.* Springer-Verlag, 1975.

[81] P. Hageman and D. M. Young. Applied Iterative Methods. Technical report, Academic Press, 1981.

[82] R. et al. Barrett. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 1994.

[83] Y. Saad. *Iterative Methods for Sparse Linear Systems*, volume 2. SIAM., 2003.

[84] L. N. Trefethen and D. Bau. Numerical Linear Algebra. In *SIAM*, 1997.

[85] S. Kaczmarz. Angenaherte Auflosung von Systemen linearer Gleichungen. *Bulle. Acad. Pol. Sci. Lett. A*, 15:355–357, 1937.

[86] C. Fletcher. Computational Techniques for Fluid Dynamics. *Springer- Verlag*, 2(ISBN 0-387-53601-9):373–395, 1991.

[87] P. W. McDonald. The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades. *ASME Paper*, 89:–, 1971.

[88] A. W. Rizzi and M. Inouye. Time Split Finite Volume Method for Three-Dimensional Blunt-Body Flows. *AIAA Journal*, 11:1478–1485, 1973.

[89] M. Farrashkhalvat and J. P. Miles. *Basic Structured Grid Generation.* Number ISBN 0 7506 5058 3. Elsevier, 2003.

[90] J. Blazek. *Computational Fluid Dynamics: principles and applications.* Elsevier, 2001.

[91] D. L. Vladimir. *A Computational Differential Geometry Approach to Grid Generation.* Number ISBN 3-540-14008-5. Springer-Verlag, 2004.

[92] D. L. Vladimir. *Grid Generation Methods.* Number ISBN 978-90-481-2911-9. Springer Science+Business Media, 2010.

[93] Q. Chang and Y. S. Wong. Recent developments in algebraic multigrids. In *Copper Mountain Conference on Iterative Methods*, volume 1, April 1992.

[94] F. Miloslav, F. Jiri, and S. Ivan. *Mathematical and Computational Methods for Compressible Flow.* Number ISBN 0-19-850588-4. Oxford Science Publications, 2003.

[95] D. Hawken. www.telusplanet.net, 2009.

[96] F. M. White. *Viscous Fluid Flow*, volume 2. McGRAW-Hill, 1991.

[97] D. Bai and A. Brandt. Local mesh refinement multilevel techniques. *SIAM J. Sci. Comput.*, 8:109–134, 1987.

Appendix A

Linear Algebra Definitions

Definition 1. Sparse matrix - a large percentage of the elements are zero.

Definition 2. Weakly diagonal dominant matrix - in magnitude, the diagonal element is at least the sum of the off-diagonal elements in the same row:

$$\sum_{j \neq i}^{n} |a_{ij}| \leq |a_{ii}| \qquad \text{for} 1 \leq i \leq n.$$

Definition 3. Define a positive definite matrix as, for all vectors $u \neq 0$, we have $u^T Au > 0$. A symmetric positive definite matrix has real and positive eigenvalues.

Definition 4. Define an M-matrix to be an $N \times N$ matrix A that is positive definite, diagonally positive, and off-diagonally non-positive.

Definition 5. Define the spectral radius of a matrix A as:

$$\rho(A) = \max |\lambda_i|, \qquad i = 1, 2, ...N,$$

where λ_i are the eigenvalues of A.

Definition 6. The discrete L^2 norm for a d-dimensional domain with uniform grid spacing h as:

$$\left\| X^h \right\|_h = \left(h^d \sum_i \left(x_i^h \right)^2 \right)^{\frac{1}{2}}.$$

תוכן עניינים

פתרון בעיות בזרימה עבר-קולית בשיטה רב-שריגית אלגברית

ע״י

שלומי שטרית

מנחים:

- **פרופ׳ אלכסנדר גלפגאט, המחלקה לזרימה ומעבר חום, הפקולטה להנדסת מכונות, אונ׳ תל-אביב.**
- **דר׳ דוד סידילקובר, המרכז למחקר גרעיני - נחל שורק.**

תקציר

מטרת עבודה זו הוא פיתוח פותרן יעיל לזרימה פוטנציאלית (זרימה דחיסה, בלתי צמיגה וחסרת ערבוליות) אשר יוכל להתמודד עם זרימה עבר-קולית, פנימית, וחיצונית עם סיבוכיות הקרובה לליניארית. החידוש בעבודה הוא ביעילות הפותרן, אשר מתקבלת על-ידי שימוש בשיטה רב-שריגית אלגברית. תוך שימוש באלגוריתם זה הודגם כי ניתן לקבל קצבי התכנסות הקטנים מסדר גודל במשטרי זרימה שונים (תת-קולי, עבר-קולי) ללא תלות בגודל הבעיה (רזולוציה). הקושי העיקרי בפתרון של שדה זרימה פוטנציאלית הוא בעובדה שהמשוואה משנה את אופייה, ממשוואה מטיפוס אליפטי בזרימה תת-קולית למשוואה היפרבולית בזרימה על-קולית. כאשר מפעילים רלקסציה נקודתית (לדוגמא, שיטת יעקובי, גאוס-זיידל) על האופרטור הדיסקרטי החד-צדדי המתקבל בזרימה על-קולית, מתקבלת סכימה לא יציבה נומרית. פתרון קושי זה הינו אחד ההישגים העיקריים בעבודה זו. פותחה רלקסציה נקודתית יציבה עבור משטר זרימה תת-קולית ועבר-קולית כאחד, ללא תלות בכיוון הזרימה הלוקאלי. האופרטור היציב מתקבל על-ידי מכפלה של האופרטור המקורי (המתקבל מדיסקרטיזציה של המשוואה עבור זרימה פוטנציאלית) באופרטור חד-צדדי מסדר ראשון. אופרטור זה מתוכנן כך שהפעלה של רלקסציה נקודתית על האופרטור המתקבל כתוצאה מהמכפלה, הינה יציבה. בנוסף מוצגים מספר נושאים הקשורים בפיתוחה והרחבתה של השיטה הרב-שריגית כגון: בניית רמות גסות, ואופרטורים המעבירים את התיקון שמתקבל בין הרמות. וזאת על-מנת להשיג יעילות מרבית בפתרון בעיות בזרימה עבר-קולית המאופיינים באי-ליניאריות קיצונית, גלי הלם, וחוסר איזוטרופיה (הנובעת מהשריג החישובי או מהמשוואה).

בנוסף פותחה שיטה משופרת לבניית רמות גסות. בשיטה הרב-שריגית אלגברית הקלאסית, בחירת קשרים חזקים וחלשים בין איברי המטריצה מתבסס על קריטריון בעל ערך קבוע, ללא תלות באופי האופרטור הדיפרנציאלי (המתאר את "אופי" הבעיה). בגישה שלנו בחירת קשרים חזקים נעשית על-ידי שימוש בקריטריון "דינאמי", המחושב במהלך הפתרון האיטרטיבי, עבור כל שורה במטריצה בנפרד. בבניית רמות גסות באמצעות קריטריון דינאמי בזרימה תת-קולית (בעיה אליפטית) ובשריגים סדורים, כאשר האי-ליניאריות אינה דומיננטית, לא נראה יתרון משמעותי על פני הקריטריון הקבוע. אך בזרימה עבר-קולית המתאפיינת באופרטור אשר אינו מקיים את התכונה M-matrix, התקבלו ביצועים טובים הרבה יותר, ובמקרים בהם מופיעים גלי-הלם, אי-ליניאריות וחוסר איזוטרופיה קיצונית, השימוש בקריטריון דינאמי הוא הכרחי מאוד לצורך התכנסות. בנוסף, פותח אופרטור אינטרפולציה משופר בעוד שהאינטרפולציה הסטנדרטית מתאימה בעיקר עבור מטריצות מסוג M-matrix. אינטרפולציה זו מכילה גם משקלים שליליים (תכונה זו הכרחית עבור בעיות שונות) ולכן מדויקת יותר ואינה מוגבלת עבור מטריצה בעלת תכונות מסוימות. בנוסף פותחה שיטת FMG בהקשר של השיטה האלגברית ככלי עזר לטיפול בבעיות לא-ליניאריות. על-ידי שימוש בשיטה זו מתקבל תנאי התחלה מדויק יותר לפתרון הבעיה על הרמה העדינה, דבר המהווה יתרון משמעותי בפתרון בעיות לא-ליניאריות אשר תנאי התחלה יכול להיות גורם מכריע בקבלת התכנסות.

בחלקה הראשון של העבודה מוצגים ביצועי השיטה הרב-שריגית האלגברית עבור מספר בעיות מודל דו-מימדיות בפתרון של המשוואה המתארת זרימה פוטנציאלית בתצורה קווזי-ליניארית. קירוב המשוואה נעשה באמצעות שיטת הפרשים סופיים בזרימה תת-קולית, קולית, ועל-קולית, בכיוונים שונים (ביחס לשריג החישובי). בנוסף הודגמה היכולת של האלגוריתם בפתרון משוואה לא-ליניארית המתארת זרימה פוטנציאלית עם הפרות קטנות (Transonic Small Disturbance).

חלקה השני של העבודה עוסק בפתרון זרימה פוטנציאלית בצורה משמרת. לצורך כך פותח פותרן דו-מימדי תוך שימוש בשריגים מבניים ומבוסס על שיטה רב-שריגית אלגברית. באמצעותו ניתן לפתור בעיות במשטרי זרימה שונים, בגיאומטריות מורכבות, וברזולוציות שונות. ביצועי האלגוריתם נבחנו במספר בעיות בזרימה חיצונית ופנימית, כאשר קירוב המשוואה נעשה בשיטת נפחים סופיים. בבעיות בהם נוצר עילוי, נעשה מימוש של תנאי קוטה-ג'וקובסקי לצורך חישוב הצירקולציה. ביצועי הפותרן מוצגים עבור מספר בעיות כגון: תעלה עם גבעה, גליל, נחיר, מנוע רקטי, ופרופילי כנף אווירודינמיים. בנוסף, ביצענו הוכחת היתכנות כאשר הסכימה הנומרית אשר מקרבת את המשוואה ההיפרבולית (זרימה על-קולית) הורחבה למקרה תלת-מימדי, וזאת תוך שימוש באותו אופרטור אשר "מייצב" את הסכימה במקרה הדו-מימדי. התקבלה רלקסציה נקודתית תלת-מימדית יציבה עבור משטרי זרימה שונים, והאופרטור שהתקבל נבדק באמצעות אנליזת יציבות של וון-נוימן.

אוניברסיטת תל – אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים ע״ש זנדמן סליינר

פתרון בעיות בזרימה עבר-קולית בשיטה רב-שריגית אלגברית

חיבור לשם קבלת תואר ״דוקטור לפילוסופיה״

שלומי שטרית

הוגש לסנאט של אוניברסיטת תל-אביב

עבודה זו נעשתה באוניברסיטת ת״א בפקולטה להנדסה בהדרכת פרופ׳ אלכסנדר גלפגאט ודר׳ דוד סידילקובר

אלול תשס״ע

אוניברסיטת תל – אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים ע״ש זנדמן סליינר

פתרון בעיות בזרימה עבר-קולית בשיטה רב-שריגית אלגברית

חיבור לשם קבלת תואר ״דוקטור לפילוסופיה״

שלומי שטרית

הוגש לסנאט של אוניברסיטת תל-אביב

אלול תשס״ע